BIT-LET

DIÁKTANÁROK



Volt valaha a hazai oktatási rendszerben, az oktatási formák közt egy olyan, amely minden másnál népszerűbb volt, s amely a diákok öntevékenységére épült. Úgy hívták ezt az oktatási formát, hogy önképzőkör. Diákok szövetkeztek némi tanári irányítással arra, hogy saját tudásukat ne csak az iskola szűkre szabott keretei közt gyarapítsák, hanem az azonos téma iránt érdeklődők összefogtak és szabadidejükből áldozva úgy igyekeztek új ismeretekre szert tenni, hogy az önképzőkör tagjai vállaltak kutatást, előadást egy-egy témában. A kör valóban körbejárt, aki ma előadott, az holnap hallgatója volt diáktársának. Népszerű volt ez a forma, mert a diákok sokkal inkább magukénak érezték az önképzőkört, mint bármilyen más formáját az oktatásnak. Az önképzőkör élesztőjévé vált a normál tanrend szerinti oktatásnak is, hiszen a diákok, akik az egyik napon az önképzőkörben magas színvonalú előadást hallgattak társuktól, s a tananyagon felüli ismeretekre tettek szert, másnap nem viselték el az unalmas, csak az előírt tananyaghoz ragaszkodó iskolai órákat sem. „Megkövetelték" tanáraiktól, hogy valóban színvonalas órákat tartsanak, s plusz ismereteket is vigyenek óráikba. Ez az önképzőkörösdi egy időben összeférhetetlen lett a magyar oktatási rendszer szigorú törvény szabta kereteivel, tanmenetével. Az önképzőkörök – tisztelet a néhány kivételes pedagógusegyéniség irányította kivételnek – az iskolákból eltűntek. Évekkel később azután megpróbálták poraiból föléleszteni, vajmi kevés sikerrel. A szűkre szabott iskolai keretek nem engedték a fölélesztést. Az önképzőkörök mindmáig vegetáltak csupán. Történt azonban, hogy megjelentek a világban a személyi számítógépek. Történt azonban, hogy a magyar oktatásügy illetékesei fölismerték, hogy a személyi számítógép olyasvalami, amit be kellene vinni a középiskolákba, s majd egy kicsit később talán még az általánosba is. A középiskolák számítógépesítése gyorsabban megoldódott, mint arra bárki is számított volna. Olyannyira, hogy maguk a pedagógusok sem akarták elhinni, nekik valóban föl kell készülni, a következő évtől oktatni kell legalábbis a diákok kicsiny csoportjainak a számítógép használatát, programozását. Történt azután, hogy a számítógépek valóban megjelentek az iskolákban. És mi történt? A diákok az első lépéseket megtették tanári irányítással, azután lévén fogékonyabbak, lévén több szabadidejük, egykettőre túlszárnyalták tanáraikat. Az iskolákban működő számítástechnikai körökben lassan általánossá vált, hogy a körök vezetői, irányítói maguk is diákok a pedagógusok legfeljebb némi felügyeletet gyakorolnak fölöttük, mintegy adminisztrálják működésüket. S ha jobban belegondolunk, ez azt jelenti, hogy a személyi számítógépek ürügyén középiskoláinkban fölfedezték az önképzőköröket. A diákok ma már olyan módon tartják kezükben saját számítástechnika-oktatásukat, hogy nem is lehetne kivenni a kezükből. Az okos tanárok nem is próbálkoznak ezzel. Inkább a tanulás módjára figyelnek, arra, hogy ne csak a legjobb 4–5 diák juthasson gépidőhöz, hogy a kezdők is haladjanak előre.

Az okos tanárok nem visszaszorítani akarják a diákok önképzési törekvéseit, hanem inkább fölhasználni hathatós gyarapodásukra.

Nemrég diákokkal beszélgettünk a számítástechnika oktatásáról. A jelenlevők többsége olvasója a BIT-LET-nek is, a Mikromagazinnak is. A srácok kaján vigyorral nyugtázták a lapunkban s laptársunkban megjelenő oktatást segítő anyagokat, oktatási terveket, koncepciócskákat. Kaján vigyorral közölték, hogy szerintük „úgy" pedig nem lehet számítástechnikát oktatni. Nem állítom, hogy zseniális, világmegváltó ötleteket, elképzeléseket soroltak a megoldásra, de egy tény: nem mondtak nagyobb balgaságokat, mint amiket szakemberek szájából is hallani olykor. Én biztattam őket, hogy írják meg a BIT-LET számára, hogy szerintük hogyan kell számítástechnikát oktatni. Sajnos az ígért kézirat még nem érkezett meg, én azonban elgondolkodtam valamin. Mi lenne, ha az oktatásügy illetékesei tudomásul vennék ezt a szerintem nagyon is előnyös helyzetet. Tudomásul vennék, hogy a diákok számítástechnika-oktatása elsősorban a diákok kezében van, s a pedagógusok számára szervezett oktatási továbbképzések, tapasztalatcserék mellett rendszeressé tennék az önképzőkörök diákvezetői számára is az efféle tapasztalatcseréket. Sőt mi több, mi lenne, ha a legjobb diákkörök vezetőiből alakított csapattal valóban megíratnák azt az oktatási segédletet, amelyben diákok adhatnának tanácsot diákoknak arról, hogy hogyan érdemes tanulni, hogyan érdemes egy önképzőkör munkáját szervezni, irányítani.

Kedves önképzőköri vezetők és tagok! Addig is, amíg az oktatásügy e kérdésben nem egyenjogúsítja Önöket, szíveskedjenek a BIT-LET hasábjait igénybe venni, s megírni, hogyan tanítanak Önök. Mit tartanak jónak s mit rossznak. Várja valamennyiük jelentkezését a szerkesztő: ***Angyalosi László***

BELÜLRŐL

ötlet

HÍROLDAL

Japán előzés

Japán ismét bizonyítja, hogy képes mindenkit megelőzni az elektronikában. Hamarosan megkezdik a kereskedelmi forgalmazását a világ leggyorsabb és legnagyobb memóriakapacitású számítógépének. Az új japán szuper számítógép 130 millió utasítást tud végrehajtani másodpercenként, memóriája pedig 256 megabyte-os. Ezzel a teljesítménnyel csupán egy készülőfélben levő amerikai számítógép hasonlítható össze, de ennek a jellemző két adata megközelítően csupán fele a japánok gépének.

Számítógépevők

A nagy számítóközpontok rémei a patkányok. Ugyanis a számítógépek tápegységeiből kisugárzó ultrahangzajok vonzzák ezeket a rágcsálókat, mivel úgy érzékelik, mintha más patkányoktól származnának az ultrahangzajok. A kártevő állatok szétrágják a gépek kábelrendszereit, belső alkotórészeit.

Észt olvasógép

Az Észt Tudományos Akadémia egyik intézetének szakemberei olyan számítógépes berendezést hoztak létre, amely például egy könyvből képes hangosan felolvasni a szöveget. A készülék úgy is alkalmazható, hogy például az írógéppel írt szöveget elolvassa és ezzel vezérelve a nyomdagépet, megvalósítja a közvetlen nyomtatást, jelentősen rövidítve ezzel az átfutási időt.

Lapos Commodore

A Commodore LCD típusú 3 font súlyú hordozható mikroszámítógép felhajtható folyékony kristályos képernyővel rendelkezik, amelynek mérete 16 sor és 80 oszlop. A gép beépített 300 Band sebességű modemmel rendelkezik, telepről vagy külső AC áramellátással működtethető. A gép rendelkezik egy 96 kbyte méretű ROM-mal. Ebben a csak olvasható memóriában szövegfeldolgozó, file-kezelő és spreadsheet programok vannak. Létezik még itt kommunikációs szoftver, emlékeztetőket rögzítő és címtárat kezelő program is.

Az LCD 32 kbyte méretű RAM memóriája teljes méretében adatok és adatállományok számára áll rendelkezésre. Az LCD-hez csatolható minden Commodore 64-hez létező periféria. A Commodore LCD-t 600 dollárnál olcsóbban fogják forgalmazni.

Okos bűnözők

Új kétes üzletág virágzik egyes fejlett elektronikával rendelkező országokban. Márkás mikroszámítógépek lemásolásával, gyors legyártásával és a feketepiacon történő árusításával, a hamisítók máris mintegy öt-hat milliárd dollár veszteséget okoztak a bejegyzett cégeknek. Szakemberek és nyomozóapparátus hiányában a hivatásos rendőri szervezetek képtelenek bármit is tenni. Az áldatlan helyzet következtében egy új szakma született, a technológiadetektíveké. Az Egyesült Államokban például egymás után alakulnak a speciális nyomozóirodák, amelyek a nagy hírű cégek megbízásából hajtóvadászatot indítottak a kalózgyártók ellen.

Pedál egér

A Versatron nevű kaliforniai cég megjelentette a piacon a kézi használatú „egér" lábpedállal megvalósított változatát. A szerkezet hasonlít a diktafonról dolgozó titkárnők lábpedáljához, szabaddá téve a kezeket a billentyűzeten végzendő műveletek számára. A gyártó azt állítja, hogy a pedálegér használata nem igényel speciális szoftvert, beilleszthető a mikroszámítógép és billentyűzet közé. Jelenleg az IBM PC-hez kapható, de rövidesen árusítják az Apple gépeihez, valamint az IBM PC XT és AT számítógépekhez. A készülék ára 225 dollár.

Koala-tábla

A Koala Pad nevű, érintéssel működő kurzormozgató tábláról közöl ismertetést a BYTE március száma.

Ez az érintéssel működő eszköz alkalmas adatok bevitelére, illetve a képernyőn megjelenő adatok kiválasztására a billentyűzet használata nélkül. Elődei: a fényceruza, a kurzor-mozgató gömbök és különféle egerek a Koala Padhoz képest drágák voltak, hiszen ennek az új terméknek az ára mindössze 125 dollár. Mérete 15x20 cm, amelyen belül az érintésérzékeny terület 27,4 négyzetcentiméter. Csatolható az Apple, az Atari és az IBM PC számítógépekhez.

Az érintésérzékeny területet ujjheggyel vagy ceruzavéggel lenyomva a képernyőn a kurzort mozgathatjuk. Két program szolgálja ki az érintőtáblát: a Micro Illustrator segítségével rajzolhatunk, majd a rajzokat tárolhatjuk és visszahívhatjuk. Az Instant Programmer's Guide olyan programrészeket jelent, amelyeket a felhasználó saját BASIC programjába illeszthet.

Kínai mikro

Mikroszámítógépek gyártására alkalmas berendezéseket, technológiát vásárol Kína a japán Nippon Electric elektronikai vállalattól. A japán cég 16 bites gépek gyártásához szállítja a szerszámokat, eszközöket és a módszereket. Együttműködnek a Nippon Electrickel olyan kódrendszer kidolgozásában is, amely lehetővé teszi a kínai írásjelek számítógépes megjelenítését. Ugyanakkor Kína és az amerikai Intel cég megállapodása értelmében több ezer összeszerelésre váró mikroszámítógép érkezik a Hongkonghoz közel eső különleges gazdasági övezet egyik elektronikai üzemébe.

Magyar szótár

Nagy munkába kezdett a Magyar Tudományos Akadémia Nyelvtudományi Intézete. Célul tűzték ki a nemzeti nyelv nagyszótárának elkészítését. A koncepció szerint a nagyszótár elsősorban szépirodalmunk szókincsét fogja tartalmazni a könyvnyomtatás kezdetétől napjainkig. A munkában nyelvészek, irodalomtörténészek, könyvtárosok, történészek és számítógépes szakemberek vesznek részt, ugyanis ezt a hatalmas munkát számítógép nélkül nehéz lenne elképzelni is.

Konferencia

Számítástechnikai konferenciát rendezett a KISZ Budapesti Bizottsága március 14-én. A SZÁMALK Szakasits Árpád úti székházában összejött ifjú szakemberek három szekcióban vitatták meg a személyi számítógépekkel és programjaikkal kapcsolatos problémá-

kat és eredményeket. Előadások hangzottak el többek között a Commodore 64 és ZX Spectrum számítógépek speciális alkalmazásairól vagy például a FORTH programozási nyelvről mint a programfejlesztés új eszközéről.
Az I. kerületi KISZ-bizottság, a SZÁMALK, az SZKI KISZ-bizottságai és a Neumann János Számítógéptudományi Társaság Ifjúsági Bizottsága által segített és támogatott konferencia jó lehetőség volt arra is, hogy a fiatalok megismerjék egymás munkáját és felhasználhassák az így közreadott tapasztalatokat.

Karóra számítógép

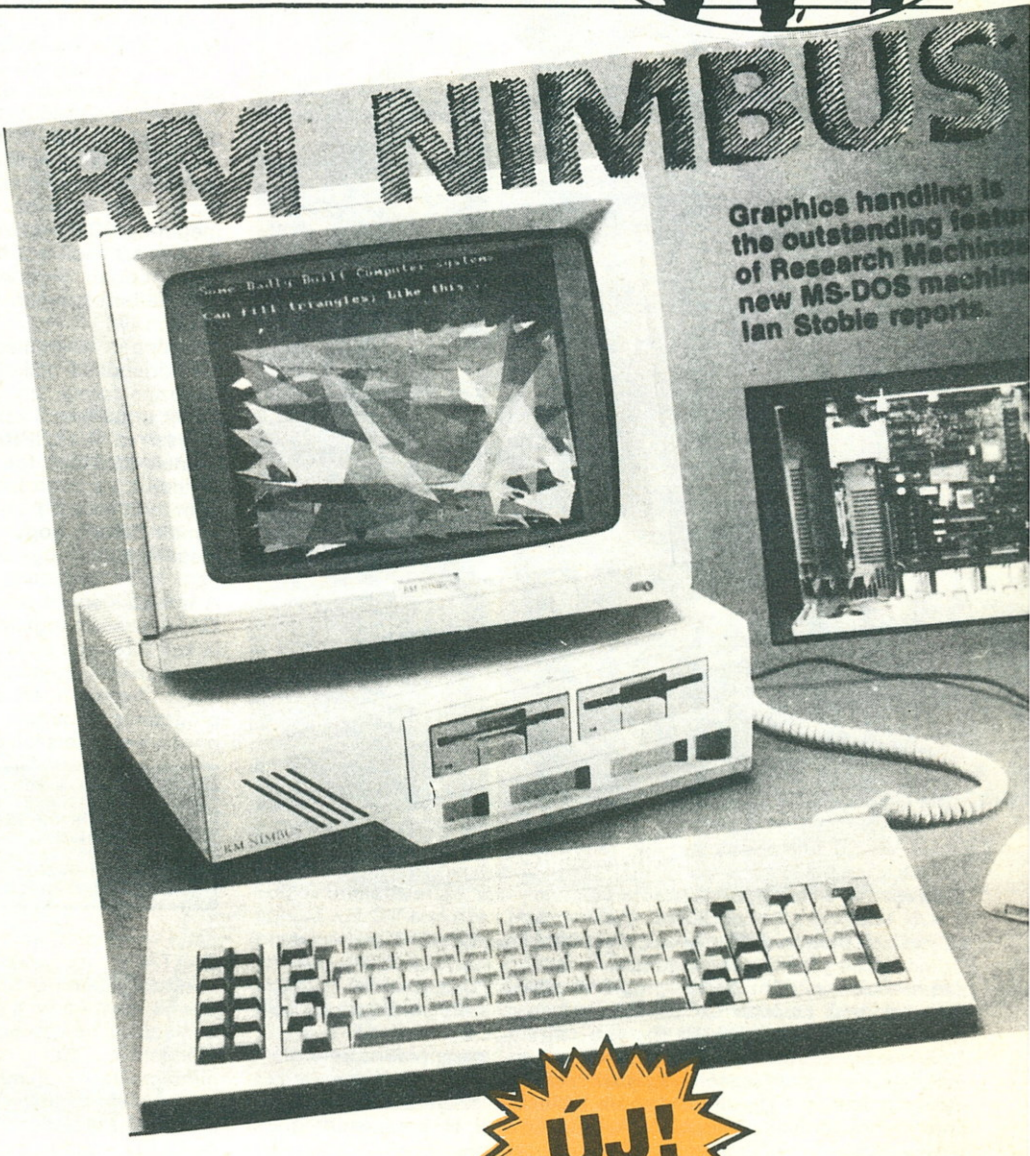
A Las Vegas-i Consumer Electronics vásáron a Seiko és az Epson olyan újabb karórákat állított ki, amelyek csatolhatók számítógéphez. A Seiko korábbi UC–2000-éhez hasonló gépet mutatott be az RC–1000-t. Ez külön billentyűzetet használ 2 kbyte méretű szöveges adat bevitelére. Ez kar-számítógép bármilyen RS–232C soros csatlakozóra illeszthető.
Az Epson új karórája egy 790 kompatibilis processzort alkalmaz egy 23 érintő-billentyűs billentyűzettel. A ROM 8 kbyte, a RAM 2 kbyte méretű. A képernyő 4 soron és 7 oszlopon képes információt megjeleníteni. A ROM-beli programok előjegyzési naptárkezelést, cím- és telefonjegyzék tárolását képesek elvégezni.

NDK újdonság

Több újdonság is érkezik a hazai piacra a Német Demokratikus Köztársaságból az idén. Ilyen például a 1715 típusú személyi számítógép, melyből közel ötven darab lesz kapható. Jövőre már mintegy ötszázat hozunk be az országba. Szintén idei újdonság lesz a 6101-es típusú, cserélhető margarétafejes, számítógéphez csatlakoztatható írógép. Az új készülékeket a MIGÉRT hozza forgalomba.

Óriás miniatűrök

Az elektronika mérföldekkel halad előre. Évente születnek szenzációszámba menő eredmények. Ilyen volt például a múlt évben sorozatgyártásra került 32 bites mikroprocesszor. Ez évben pedig a 256 kilobites memóriachipek tömeggyártásának beindulása a nagy esemény. Ugyanakkor tudjuk, hogy már elkészült az egy megabites tárolóchip, és várható, hogy a jövő év szenzációja éppen ennek a forgalomba kerülése lesz.



ÚJ!

A Research Machines nevű angliai cég sokat váratott az új 16 bites személyi számítógépének megjelentetésére. A személyi számítógép operációs rendszere az MS–DOS, a gép „majdnem IBM kompatibilis". Alapkiépítése monitor nélkül, 1220 angol fontba kerül, ami meglehetősen versenyképes árat jelent. A termék versenyképességét a tervezők azzal igyekeztek fokozni, hogy dedikált grafikus társ-processzorral látták el a gépet. Ezzel a módszerrel sikerült elérni, hogy a gép grafikus lehetőségei kiemelkedőek és gyorsak.
A gép műszaki adatai:
Központi egység: Intel 80186, grafikus célprocesszor, opcionálisan egy 8087-es matematikai processzor.
Memória: 192 K (ebből 64 K dedikáltan grafikus célokra) 1 Mbyte-ra terjeszthető ki. A ROM mérete 64 K, rendszer- és grafikus rutinokat tartalmaz.
Billentyűzet: különálló, IBM PC billentyűzet különválasztott numerikus kurzor billentyűzet résszel. Az „egér" opcionális.
Hang: 8910-es „zene-chip" három csatornán, hat oktáv, digitalizált hang output lehetőség egy külön Oki csippel.
Háttértárak: négy különböző változat létezik a hálózatokba való lemez nélküli változattól a maximális kiépítésig, két 3.5 inch-es Sony mikrofloppy (720 KByte/floppy) plusz egy 20 Mbyte-os merevlemezes tároló.
Képernyő: fekete-fehér vagy színes. 25 sorx40 vagy 80 oszlop. 250 vízszintesx320 vagy 640 függőleges képpont. A szöveg és az ábrák keverhetőek a képernyőn.
A gépet a gyorsaság és a jó grafikus lehetőségek jellemzik, ezért az oktatási, műszaki és a tudományos alkalmazásokkal jellemehető piaci szegmenseknek szánják.
A gép sebességét 8 Basic rutin 1000-szeres végrehajtásával tesztelték. Az összehasonlító adatok:

Géptípus:	**Átlagos végrehajtási idő**
RM Nimbus (80186):	6.2
IBM PC/AT (80286):	6.8
Acorn BBC (6502):	14.8
IBM PC (8088):	16.8
RM 4802 (Z80):	19.6

VALLATÓ

VC 20



Mivel VC 20 tulajdonos vagyok, lehetetlen, hogy tollat ne ragadjak vallatásának hatására.

Először is a hanggal kapcsolatban van egy észrevételem. A Junoszty televízió áthangolásával kiválóan szól azon is a VC 20. Sajnos ehhez bele kell nyúlni a televízióba (szakembernek), mivel azt nem Commodore gépekhez tervezték.

Másodszor a gépi kódú programozásával kapcsolatos felületes vallatásukhoz kívánok szólni. Ez ügyben összehasonlítom a Z/80 gépi kódú programozásával.

1. Eleve egyszerűsíti a helyzetet, hogy a processzornak a veremtármutatón és programszámlálón kívül csak egy akkumulátora és két indexregisztere (XJ) van, továbbá a feltételregiszter, szemben a Z/80 regiszterkészletének tömegével (A, F, B, C, D, E, H, L a másodregiszterek IX, IY).

Ami azonban nem teszi ügyetlenebbé a gépet, mivel a használható zéró-lapos címzési forma lehetővé teszi, hogy mindegyik 0-lapos memóriacímet REGISZTERKÉNT kezeljük, és így tulajdonképpen 256 (!) plusz regiszter birtokosainak érezhetjük magunkat!

2. A processzornak jóval kevesebb az utasításkészlete mint a Z/80 processzornak (152 db ismert utasítás + 32 db ismeretlen, szemben a Z/80 több mint 700 utasításával), ennélfogva könnyebben áttekinthető és megjegyezhető.

Sőt! Míg a Z/80 műveletet csak az akkumulátorral, a regiszterekkel a HL, IX, IY regiszterpárokkal képes végezni, a mi processzorunk a memória összes byte-jával képes műveletet végezni. Például ha a Z/80-nál az akkumulátorhoz hozzá akarjuk adni egy cím tartalmát, akkor először betöltjük a címet a HL regiszterpárba, majd következő utasításban hozzáadatjuk az akkumulátorhoz a HL által mutatott cím tartalmát. Mi azonban egyetlen utasításban hozzá tudjuk adni az akkumulátorhoz bármelyik cím tartalmát

Z/80 LD HL, cím 6502: ADC cím
ADC A,(HL)

Így a Z/80-nál szükségessé válik a regiszterekbe töltés – és a regiszterek közötti csereutasítások nagy számú megnövelése. Vagy itt vannak a Z/80 16 bites regiszterpárjai, amit mi kiválóan helyettesíthetünk egy zérólapos címzéssel.

Például hasonlítsuk össze az akkumulátor tartalmát az IX regiszterpár és egy d eltolódás által mutatott cím tartalmával a Z/80 esetében:

CP (IX+d) (3 byte hosszú)

És hasonlítsuk össze a 6502 esetében:

CMP (z)+y (2 byte hosszú)

ahol z egy két byte-os cím első byte-jára mutat a zérólapon és az eltolódást az Y regiszter tartalma jelenti. Sőt, míg a Z/80-nál csak IX-et tudjuk módosítani a zérólapos cím mindkét byte-ját, és az eltolódást jelző Y regisztert is!

Így valójában úgy vetődik fel a kérdés, hogy a kétféle processzor közül melyik tud több 16 bites regiszterpárt használni. A Z/80 használja a BC, DE, HL, IX, IY regiszterpárt, a 6502 pedig felhasználhatja a zérólap 256 byte-ját „regiszterpárként"! (Azaz a 128-at.)

Nem csoda, ha a Z/80 memóriakezelése szegény, jócskán ki kell bővíteniük utasításkészletüket regiszterekbe töltő és memóriacímre töltő utasításokkal, ezért az a természetes LD utasítás, annyira, hogy még tömbös töltőutasítás is szükségessé vált!

3. Megírtam egy DISASSEMBLER és egy ASSEMBLER programot VC 20-ra gépi kódban. Segítségükkel ezerszeresen egyszerűsödött maguknak a kódoknak a fordítása is. Mivel a programok kb 1–1 kbyte-osok, ezért legalább 1 kbyte-os gépi kódú anyagot tudok egyszerre megírni a VC20-ra!

4. Aki azt mondta, hogy a gépet memóriabővítés nélkül lehetetlen gépi kódban programozni, az nyilvánvalóan nem is értett hozzá. Amióta birtokomban van, memóriabővítés nélkül programozom gépi kódban.

Amiképpen lehet még az 1 K-s ZX81-et is gépi kódban programozni memóriabővítő nélkül, annál inkább lehet a VC 20-at. Elterjedt programozási forma, hogy gépi kódú anyagunkat egy BASIC sorban a REM utasításban tároljuk, kikeressük, hogy melyik memóriacímre helyezi az INTERPRETER, és onnan hívjuk meg gépi kódú szubrutinunkat. Másik forma, hogy DATA-ban tároljuk és programfutáskor kiküldjük az adatokat egy letiltott területre, ahol az INTERPRETER nem nyúl hozzá. Az előző formátum hátránya, hogy ki kell keresnünk a memóriából, hogy az INTERPRETER hová pakolta sorunkat, és így megállapítani, hogy melyik címet hívjuk meg, ami már kettőnél több gépi kódú szubrutin írásánál komoly gonddal és vesződséggel jár. A második formátum hátránya pedig a nagy helyfoglalás. Egyszer a DATA soraink foglalnak le a memóriában területet, másodszor pedig le kell tiltanunk egyéb szabad területet is, ahová küldjük a gépi kódú anyagunkat programfuttatáskor. A VC 20-nál mindkét formátum adott. Azonban én ajánlok két egyszerűbb és könnyen kezelhető programot, amellyel gépi kódú anyagunkat kazettára vihetjük és onnan pedig tetszőleges helyre beírhatjuk.

```
1000 OPEN1,1,1:REM KAZETTAS CSATORNA MEGNYITASA
1001 INPUTA:IFA<0THEN1005:REM ADATOK KERESE,MIG NEGATIV ADAT NEM JON
1002 K=K+A:N=N+1:REM K=ADATOK OSSZEGE,N=ADATOK SZAMA
1003 PRINT#1,A:REM KAZETTARA IRAS
1004 GOTO1001:REM FOLYTATAS
1005 CLOSE1:PRINTK;N:REM KAZETTAS CSATORNA LEZARASA,K;N KIIRASA
1006 NEW:REM A PROGRAM KITORLESE A MEMORIABOL
```

Ezt a rövid programot futtatva először a „PRESS RECORD & PLAY ON TAPE" üzenet jelenik meg. Nyomjuk meg mindkét billentyűt a kazettás magnón. Mikor megáll a magnó, a képernyőn a kérdőjel jelenik meg, várva a gépi kódú adatokat. Azokat szép sorjában adjuk be, legutoljára egy nullánál kisebb számot. Ha sok adatot viszünk be, időnként beindul a magnó, és kazettára kerül az anyagunk. Ilyenkor mindig várjuk meg míg a kérdőjel megjelenik. Amikor beadtuk a legvégét jelző negatív számot, akkor két szám jelenik meg: az első egy ellenőrző szám, a gépi kódú anyagunk adatainak összege, a második pedig az adataink száma. Végül a programunk önmagát kitörli a memóriából.

Ha most azt akarjuk, hogy a kazettára mentett gépi kódú anyagunkat a gépbe vigyük, üssük be a következő programot.

```
2000 INPUT"MELYIK CIMTOL";P
2001 POKE56,P/256:POKE55,PAND255:CLR
2002 OPEN1,1,0
2003 P=PEEK(55)+256*PEEK(56)
2004 INPUT#1,A
2005 POKEP,A
2006 K=K+A:N=N+1:P=P+1
2007 IFST=0THEN2004
2008 CLOSE1:PRINTK;N
2009 NEW
```

A szerkesztő azért van,

hogy a lap olyan legyen,

mint amilyenek az olvasói!

2000: sor: Tudnunk kell, hogy 3583 byte áll rendelkezésünkre, 4096–7679 címeken. Tehát kezdő címünk bárhol elhelyezkedhet ezen a területen, természetesen hagyjunk helyet BASIC-programunknak is, amely 4096-os címen kezdődik.
2001: sor: Letiltja az INTERPRETER számára azt a területet, ahova anyagunkat helyezni szeretnénk.
2002: sor: Kéri, hogy kapcsoljuk be a PLAY gombbal, megindítja a magnót és beáll a kazettán levő adataink elejére.
2003: sor: A CLR utasítás törölte P értéket, de az 55–56 címen az tárolva van.
2004–2006: Beküldi összes adatunkat a meghatározott címtől kezdődően kazettáról a memóriába, számolja összegüket és darabszámukat. Ezt mindaddig csinálja, míg az adatok vége jelzést nem kapja kazettáról.
2008: Leállítja a magnót, kiírja az ellenőrzőszámot és adataink számát, itt ellenőrizhetjük, hogy rendben ment-e minden.
2009: ez törli BASIC-programunkat a memóriából.
Most tehát betöltöttük gépi kódú anyagunkat a gépbe, tudjuk hol keressük, a többi terület teljesen szabad, betölthetjük oda azt a BASIC-programot, amivel gépi kódú szubrutinjainkat használni akarjuk. Maga a töltőprogram pedig semmiféle helyet nem foglal el, mivel kitörlődött.

5. Dokumentációval kapcsolatos észrevételem. Valóban csapnivalóan sovány információval látták el a gépet. Ez ügyben felajánlom saját információimat, amiket szereztem a gépről, mivel lefordítottam magamnak a teljes INTERPRETER anyagát.

Végezetül, véleményem szerint a VC 20 gépi kódú programozása nemcsak egyszerű, könnyen elsajátítható, de maga a nyelvezete is rendkívül ötletes. Gépi kódú programjaink némiképp gyorsabbak is lehetnek azáltal, hogy akkumulátorunk közvetlen kapcsolatba kerülhet a memória összes byte-jával, és így nem kell azokat állandóan töltögetni ide vagy oda.

Tóth Kornél 1083 Leonardo da Vinci u. 29. fsz. 8.

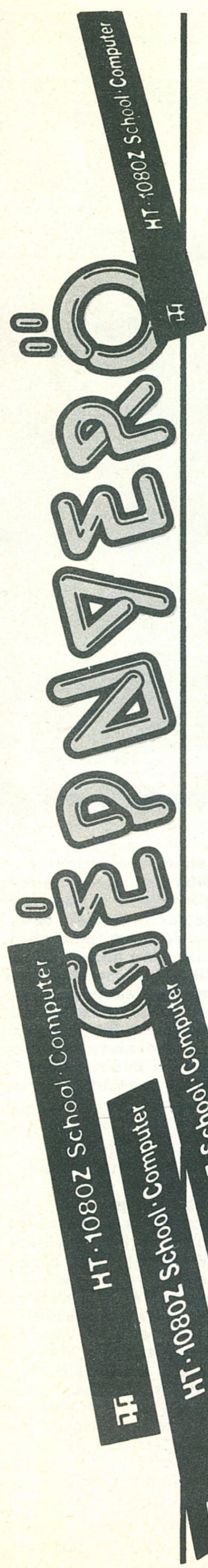
PRIMO



Olvastam a PRIMO vallatását a január 31-i számban. Elsősorban ehhez szólnék hozzá. Összességében én a géppel meg vagyok elégedve (PRIMO A–64), azonban a hozzá kapott tápegységgel egyáltalán nem. Ahogy Amtmann Árpád írta: „Bírja a 72 órás kikapcsolás nélküli üzemet!" Hát ehhez én azt mondanám; a gép igen, de a tápegység egyáltalán nem. A tápegység egy hét (kíméletes) használat után bemondta az unalmast, ki kellett cserélni. A második egység eddig még ilyen szempontból jó, de a géphez jövő csatlakozója olyan rosszul van beforrasztva, hogy ahogy mozgatom a vezetéket, a kép borzasztó sokat változik, sőt néha tiszta fehér a kép, még reszetre sem áll vissza normál üzemmódba, s ha nem mozdítom meg ismét a vezetéket, akkor örökké csak a nagy fehér semmit látom. Ha szerencsém van, akkor a vezeték olyan állapotban van, hogy a kép nagyon-nagyon szép (3 hónap alatt kétszer).

Editálás **Négyes osztályzatot kapott a gép, holott szerintem a PRIMO-nak ez a leggyengébb pontja. Néha, ha megmakacsolja magát, nem engedi átjavítani a hibás sort. Illetve átengedi, de vissza is írja. Ekkor általában az sem segít, ha újraírom a sort. Ilyenkor a legbiztosabb módszer, ha megkeresem, hol van a tárban az a bizonyos hiba és POKE-kal javítom. Kissé hosszadalmas. A másik ok, amiért írok, a következő: a Vallatóban azt írták, hogy elkészült vagy előkészületben van a gépkönyv, szoftvertérkép, assembler, disassembler, Forth, Pascal. Én igényt tartanék ezekre. Ha kell, kazettát is küldök, s postán a programnak és a többinek az árát. Visszatérve magára a gépre, valóban elég jó teljesítményű kisgép, eltekintve ettől a pár hibától (EDIT). Valóban hiányzik belőle a DRAW és a FLASH utasítás, de ezeket kis ügyességgel a PRIMO-n könnyű pótolni, még BASIC-ben is. Valóban jó tulajdonsága a gépnek, hogy a kulcsszavakat rövidített formában (különböző CHRS-ben használatos jelekkel) is elfogadja. Csak ugye néha egyszerre három billentyűt úgy fogni, hogy a lehető legnagyobb felülettel érjünk hozzá (vagy négyet)! SHIFT+ ↓ +egy billentyű, vagy SHIFT+ ↓ +CTR+ billentyű.**

egy PRIMO-tulajdonos: *Fehér Csaba,* 8360 Keszthely, Fürst S. u. 10.



PÁLYÁZATI ÉRTÉKELÉS

HT-nyerő pályázatunkban utolsóként az oktatóprogramokat értékeltük. Egy sajnálatos félreérthető mondat került a pályázat kiírásába. A „BASIC oktatóprogramot kell írni" feltételt sokan úgy értelmezték, hogy BASIC nyelven írott oktatóprogramot kérünk, holott elképzelésünk szerint a BASIC nyelvet oktató programot szerettünk volna. Mivel azonban a hiba valóban a mi készülékünkben, azaz a kiírásban volt, elfogadtunk minden oktatóprogramot.
A pályázatba az elején bekapcsolódó ***72 szakkör közül végül is 26 jutott el a finisig,*** s küldte be ezt a valóban komoly munkát igénylő programot. A 26 csapat közül a legjobb 288 pontot, a leggyengébb 191 pontot ért el összesen.

Néhány szót az oktatóprogramokról:
Témájukat tekintve sokfélék voltak a programok. A félreértés ellenére sokan írtak valóban a BASIC nyelvet oktató programot, volt biológiát, nyelveket – például németet, olaszt – oktató program, volt fizikai szimulációs program, sőt sakkoktató program is. Közölni ezek közül egyiket sem áll szándékunkban, egyfelől, mert terjedelmük meglehetősen nagy, másrészt pedig a legjobbak olyannyira jól sikerültek, hogy a TII szeretné ezeket az oktatási programpályázatban is értékelni, s valószínűleg egyiket-másikat meg is venné. Mivel ahhoz, hogy az oktatási programpályázatban ezek a programok értékelhetők legyenek, bizonyos kiegészítéseket a programok készítőinek be kell adni, kérjük, hogy a pályázók – ha ezen követelményeknek eleget akarnak tenni – feltétlenül keressék meg a Tudományszervezési és Informatikai Intézetet levélben.
(Címünk: Budapest Pf. 454. 1372)

Az iskolák, amelyeknek pályázatairól szó van, a következők:
Katona József Gimnázium – Kecskemét ● Berze Nagy János Gimnázium – Gyöngyös ● Berzsenyi Dániel Gimnázium – Budapest (a kémiai programot író szakkör) ● Piarista Gimnázium A, B, C, D szakkörei – Budapest ● Földes Ferenc Gimnázium – Miskolc ● 600. sz. Ipari Szakmunkásképző Intézet – Szeged ● Széchenyi István Gimnázium – Sopron ● Svetits Katolikus Gimnázium – Debrecen ● Varga Katalin Gimnázium – Budapest
Az oktatóprogramok készítésénél természetesen azoknak a szakköröknek sikerült igazán színvonalas munkát produkálniuk, amelyek tagjai együtt dolgoztak az iskola valamelyik szaktanárával. Értelemszerű, hogy egy biológiaoktató program elkészítéséhez nem elég programozni tudni, de a biológiában is bő ismeretekkel kell rendelkezni. A szaktanári segítséget versenykiírásunk sem zárta ki, s értelemszerű volt, hogy erre szükség van. Sajnos a pályázatok egy részének értékét éppen a tartalmi, didaktikai hiányosságok rontották le, nyilvánvaló volt, hogy ezek a pályázók vagy nem vettek igénybe szaktanári segítséget, vagy nem találtak jó partnerre. Olyan programot is találtunk, amelynek megnézésénél az volt a véleményünk, hogy azt a témát, amit a program oktatna, érdemesebben és eredményesebben lehet a hagyományos módszerekkel, eszközökkel oktatni. Jó tanulság volt, hiszen kiderült, hogy valóban nem érdemes minden oktatási folyamatba bevonni a számítógépet.
A pályázat egészére jellemző volt, hogy a beküldött programokhoz sokan még azt a minimális dokumentációt sem készítették el, amely ahhoz szükséges, hogy valaki gond nélkül tudja kezelni, használni a programot. Tudjuk, hogy az írás nem a számítógépesek kenyere, de azt egyszerűen tudomásul kell venni, hogy minimális dokumentáció nélkül nem adhat ki programot a kezéből az ember. Épp ezért már jó előre figyelmeztetjük pályázóinkat, hogy újabb pályázatainkban az ilyen minimális dokumentáció nélkül érkező programokkal nem fogunk bíbelődni, egyszerűen értékelés nélkül adunk rá nulla pontot. Tessék ugyanis elképzelni azt, amikor a program kezelését a programlistából kell kisilabizálni. Nem kellemes, ugye? Pályázatunk sikere egyébként várakozáson felüli volt. Vonatkozik ez a résztvevő szakkörök számára, hiszen itt valóban komoly munkát igénylő feladatokat kellett megoldani, de vonatkozik éppígy a megoldások színvonalára is.
Gondolkodunk rajta például, hogy a liftes feladat megoldásai közül a legjobbakat továbbítjuk majd megfelelő szakemberekhez tanulmányozásra, merthogy a működő liftek nagy részének ezeknél lényegesen rosszabbak a programjai, ez biztos. A sikeren fölbuzdulva elhatároztuk, hogy újabb pályázatokat indítunk majd a szakköröknek. A második, rövidebb és kisebb munkát igénylő pályázatot már BIT-LET-ünknek ebben a számában elindítjuk. Akit érdekel, lapozzon az utolsó, 32. oldalunkra.
De azt is elhatároztuk, hogy szeptemberben a lezajlott pályázathoz hasonló méretű, nagyobb akciót is indítunk majd, amelynek valószínűleg ismét egy gép lesz a nyereménye.
A pályázat színvonala s a TII lehetőségei ragadtattak bennünket arra is, hogy a pályázat beígért első díja mellett további két komoly díjat adunk ki.
Reméljük, hogy a pályázaton részt vevők hozzánk hasonlóan jó emlékeket őriznek meg a HT-nyerőről. Nekünk sok munkánk volt vele, de élveztük:

A pályázat értékelői:

Mihályfi János, a Tudományszervezési és Informatikai Intézet munkatársa
Király Zoltán, a BIT-LET gépnyerő rovatvezetője

A Tudományszervezési és Informatikai Intézet és a BIT-LET közös HT-gépnyerő pályázatának végeredménye. Az első tíz helyezettet közöljük.

1. ***BERZE NAGY JÁNOS GIMNÁZIUM – GYÖNGYÖS***
2. ***BERZSENYI DÁNIEL GIMNÁZIUM haladó szakköre – BUDAPEST***
3. ***SVETITS KATOLIKUS GIMNÁZIUM – DEBRECEN***
4. ***PIARISTA GIMNÁZIUM „A" SZAKKÖRE – BUDAPEST***
5. ***SZÉCHENYI ISTVÁN GIMNÁZIUM – SOPRON***
6. ***600. sz. IPARI SZAKMUNKÁSKÉPZŐ INTÉZET – SZEGED***
7. ***KATONA JÓZSEF GIMNÁZIUM – KECSKEMÉT***
8. ***PIARISTA GIMNÁZIUM „B" SZAKKÖRE – BUDAPEST***
9. ***FÖLDES FERENC GIMNÁZIUM – MISKOLC***
10. ***VÖRÖSMARTY MIHÁLY GIMNÁZIUM – ÉRD***

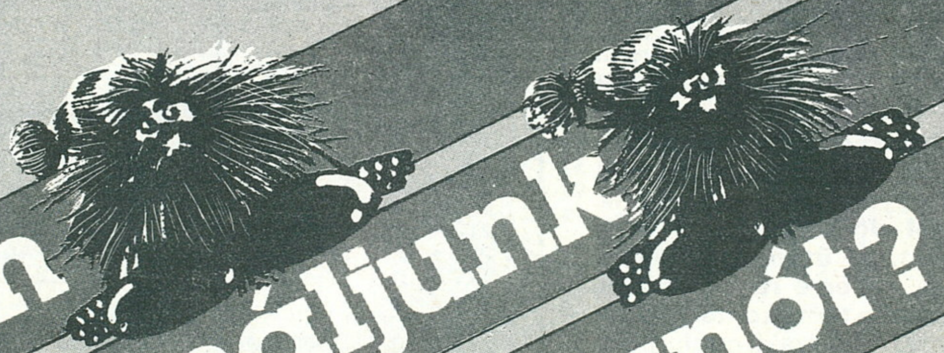
A pályázat első díja: 1 db HT 1080Z típusú 64 kbyte-os számítógép + 1 db Super Star 20 típusú televízió

A pályázat második díja: 1 db PRIMO A32 típusú számítógép PSA–01 tápegységgel + 1 db televízió

A pályázat harmadik díja: 3000 forint értékű TII oktatóprogramok vásárlására jogosító utalvány + 2000 forint értékű könyvvásárlási utalvány

Az első tíz helyezett szakkör oklevélben részesül. A jutalmak átvételével kapcsolatos információkat az iskolák levélben kapják meg!

PROGRAMAJÁNLAT C 64-HEZ



Hogyan csináljunk manót?

Még mielőtt bárkinek is pikáns gondolatai támadnának a cím elolvasása után, le szeretném szögezni, hogy az a manó nem az a manó. Manó (angolul sprite ejtsd: szprájt) a Commodore 64 egyik grafikai lehetősége. Az általunk tervezett sprite-ot (a továbbiakban nevezzük csak így) tetszés szerint tudjuk mozgatni, színezni, ütköztetni, nagyítani stb. A játékprogramokban mozgó figurák is sprite-ok. A sprite grafika ismeretében már csak egy lépés a számítógépes animáció.

Egy sprite egy 24x21-es mátrixban épül fel. 24 egység vízszintesen, vagyis 3 karakternyi. Ez azt jelenti, hogy 3 byte értéket kell megadnunk 1 sorhoz. 21-hez 63 darab 0–255 közti számot kell beütnünk. Azt már sokan tudják, hogy hogyan lehet bitekből byte-okra átszámítani, de azok kedvéért, akik még nem tudják, itt most röviden leírom.

128	64	32	16	8	4	2	1
1	0	0	1	1	0	0	1

A biteket jobbról balra számozzuk meg a 2 hatványaival; Ahol a sprite egy pontja világít, ott annak a bitnek az értéke 1; Ha egy vagy több bit be van kapcsolva (azaz 1 az értékük), akkor a byte értéke a fölötte levő számok öszszege. Példánkban ez így néz ki: 128+16+8+1=153. Ez a byte értéke. Ez sajnos hosszú, fáradságos munka, és ráadásul sok számolást igényel. A cikk végén található program ezt a munkát teszi kényelmesebbé. (De erről majd később.)

A kérdést joggal teheti fel az olvasó, jó-jó, megvannak a byte-ok, de hova töltsük ezeket? A memória tetszőleges területeire tölthetünk, kivéve a rendszerterületeket. A sprite-ok helyére blokkokkal hivatkozhatunk. Ha a 11. blokkba szeretnénk a sprite-ot elhelyezni, akkor a 11x64 (704)-es címtől kell a sprite-ot betölteni. A használható blokkok 11, 13–15, 32–63, 128–255. Ez azt jelenti, hogy 162 sprite fér el a memóriában. De a képernyőn egyszerre csak 8 darab lehet. Egyesek már hallottak arról is, hogy egyszerre 16 sprite mozgott a képernyőn (hardverátalakítás nélkül). Elvileg ez lehetetlen, de egy kis csalással bárki meg tudja csinálni, hogy a látszat az legyen, hogy az 16 sprite.

A képernyőn levő sprite-okra a 2040–2047-es címeken tudunk hivatkozni, természetesen a blokkszámukkal. Pl.: azt szeretném, hogy az első sprite a 13-as, a második pedig a 176-os blokkból kerüljön ki, akkor ezt kell beírnom: POKE 2040,13:POKE 2041,176. A megjelenítést pedig az 53248 és 53263 közti megfelelő címek átírásával érhetjük el. Egy sprite helyét két byte adja. Az első az x a második az y koordináta. Ez így túl egyszerűnek tűnik, de nem kell megijedni, mert még két cím ismerete szükséges ahhoz, hogy egy sprite-ot tologatni tudjunk. Az egyik az 53269-es. Itt azt adjuk meg a megfelelő bit értékének „bepókolásával", hogy mely vagy melyik sprite-ok jelenjenek meg a képernyőn. (Sorszámuk szerint.) A bit beírása a cikk elején leírtak alapján történik. Néha előfordul az is, hogy a sprite pozíciója nagyobb, mint 255. Ekkor az 53264-es byte-nak azt a bitjét kell átírni, amelyik sprite-koordinátája nagyobb, mint 255. Az x koordinátát jelző byte értékét most újra 0-tól írhatjuk.

Összefoglalva, ha például az első sprite-ot el akarom helyezni a

100,100 koordinátájú pontban, akkor ezt pötyögöm be: POKE 53269,1:POKE 53248,100:POKE 53249,100. A képernyőn levő 7 sprite színét 53287–53294 közti 7 címmel adhatjuk meg. (A 0. sprite 53287, a 7. sprite 53294). Ugyanúgy, mint a keret és az alap beállításánál itt is a színkódot kell megadni.

Az utolsó dolog, ami még fontos lehet, az 53277-es byte megfelelő bitjének átállításával a sprite-ot x irányba kétszeresére nagyíthatjuk. Az 53271-es byte pedig az y irányú nagyításról gondoskodik. Ezen kívül előállíthatunk 3 színű sprite-okat, az ütközéseket vizsgálhatjuk, de erre itt most nem térek ki. Akit a téma komolyabban érdekel, az a szakirodalomban biztosan megtalálja, amire kíváncsi.

A program:

Ez a program a sprite-tervezés kényelmetlen feladatát teszi könnyeddé és játékossá. Egy 21x24-es mezőben tudunk egy pontot (kurzort) mozgatni. A következő billentyűk segítenek ebben

Q – fel. A – le. O – balra. P – jobbra. A pontot a -val tudjuk rögzíteni. A *-gal letörölhetjük. A ↑ -lal az egész sprite-tól megszabadulunk. A gép a tervezés megkezdésekor az x, y irányú nagyításra kíváncsi. 0-val vagy 1-gyel válaszolhatunk a kérdésekre. A multi c kérdésre, ha nem 3 színű sprite-ot szeretnénk készíteni 0-t írjunk. A színt pedig a szín kódjával adjuk meg. Ha úgy döntöttünk, hogy a tervezéssel kész vagyunk, az S betűt üssük le. Ez a SAVE-et jelenti. Kétféleképpen lehet kimenteni a „remekműveket''. Az egyik a program üzemmód, a sprite adatait BASIC programmá alakítja át, és azt írja ki a képernyőre a kívánt sorszámtól, és a sprite-tervező programot végképp kitörli a memóriájából (jó tanács: csak ha felvettük a programot, azután próbáljuk ki ezt a részt).

Nekünk már csak az a dolgunk, hogy felmenjünk a kurzorral (az eredetivel) a gép által készített program első sorára, és addig nyomjuk a RETURN-t (de ne olyan erősen, ahogy a TV-BASIC-ben teszik), amíg a képernyő aljáig nem érünk. Ekkor BASIC-programként vehetjük fel, és már

```
1 REM ***************************
2 REM *                         *
3 REM * SPRITE TERVEZO PROGRAM *
4 REM *   (C) SZABO G.A. 1985   *
5 REM *                         *
6 REM ***************************
90 DIM A(62):FORI=0TO62:POKE832+I,0:NEXT
100 POKE53280,0:POKE53281,242:PRINT"■■";
110 PRINT"┌──────────■SPRITE TERVEZO■──────────┐";
120 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
130 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
140 PRINT"| E PROGRAM SEGITSEGEVEL TETSZOLEGES   |";
150 PRINT"| FIGURAKAT TERVEZHETSZ EGY 24*20 -AS  |";
160 PRINT"| MEZOBEN UN. SPRITEOT                 |";
170 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
180 PRINT"| ■KEZELES:■                            |";
190 PRINT"|               KURZOR FEL.....Q       |";
200 PRINT"|               KURZOR LE......A       |";
210 PRINT"|               KURZOR BALRA...O       |";
220 PRINT"|               KURZOR JOBBRA..P       |";
230 PRINT"|               KURZOR MARAD...@       |";
240 PRINT"|               UJRAKEZDES.....↑       |";
250 PRINT"|               SAVE...........S       |";
260 PRINT"|               TORLES.........*       |";
270 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
280 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
290 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
300 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
310 PRINT"|              INDITAS:■F1■             |";
320 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
330 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
340 PRINT"└───■(C)COPYRIGHT CITY SOFTWARE 1985■───┘";
350 GETT$:IFT$=""THEN350
360 IFT$="■"THENGOTO 400
370 GOTO 350
400 PRINT"■"
410 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■NAGYIT X";NX
415 IF NX<0ORNX>1THENPRINT"■" :GOTO410
420 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■NAGYIT Y";NY
425 IF NY<0ORNY>1THENPRINT"■";:GOTO420
430 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■MULTI  C";MC
435 IF MC<0ORMC>1THENPRINT"■";:GOTO430
450 IF MC THEN GOTO 500
460 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■SZINE   ";SZ
470 IF SZ<0ORSZ>255THENPRINT"■";:GOTO 460
480 GOTO 600
500 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■SZIN 1  ";S1
510 IF S1<0ORS1>255THENPRINT"■";:GOTO 500
520 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■SZIN 2  ";S2
530 IF S2<0ORS2>255THENPRINT"■";:GOTO 520
540 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■SZIN 3  ";SZ
550 IF SZ<0ORSZ>255THENPRINT"■";:GOTO 540
600 POKE53271,NY:POKE53277,NX:POKE53287,SZ:POKE53285,S1:POKE53286,S2
610 POKE53276,MC*9:PRINT"■"
620 FORI=0TO20:PRINT"  ■                              ":NEXT:
623 PRINT"■┌──────────────────────────────────────┐";
624 FORI=1TO22:
625 PRINT"|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|";
626 NEXT
627 PRINT"└───■(C)COPYRIGHT CITY SOFTWARE 1985■───┘";
630 A$="*":Y$="■■■■■■■■■■■■■■■■■■■■■■■■■":
640 X$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■":Y=10:X=10
645 POKE53248,255:POKE53249,180:POKE53269,9:POKE2040,13
660 GETT$:IFT$=""THEN660
670 IF T$="Q"ANDY>1THENGOSUB1000
680 IF T$="A"ANDY<22THENGOSUB1100
690 IF T$="O"ANDX>2THENGOSUB1200
695 IFT$="↑"THENFORI=0TO62:POKE832+I,0:NEXT:GOTO 400
697 IF T$="S" THEN GOTO 7000
700 IF T$="P"ANDX<25THENGOSUB1300
710 IF T$="@"THENGOSUB1400
720 IF T$="*"THENGOSUB1500
800 GOTO 660
1000 V=160
1010 A=PEEK(Y*40+X+1024):IF A=81THENV=209
1020 PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X);"■ ";"■";CHR$(V )
1025 IF PEEK((Y-1)*40+X+1024)=209THEN 1035
1030 Y=Y-1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)".";:
1033 RETURN
1035 Y=Y-1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)"●";
1040 RETURN
1100 V=160
```

csak egy betöltő részt kell elé írnunk. A másik fajta felvétel esetén úgy vehetjük fel a programot magnóra, hogy az adatok egy A tömbben helyezkednek el, és mint adat kerülnek rögzítésre, ezért a betöltésnél is így kell behívni.

A működés:

100–340 Főcím, kezelés leírása
400–500 A sprite adatait kéri és ellenőrzi a program
600–610 A kapott adatokat a sprite-nak beállítja
660–800 A billentyűzetet ellenőrzi
1000–1340 A lépéseket hajtja végre
1400–1470 „Bepókolja" a „bepókolni" valókat
1500–1570 „kipótolja" a „kipótolni" valókat
7000–7230 Save rutin képernyője
8000–8030 Programkénti mentés
8100–8150 Adatkénti mentés

A fontosabb változók:

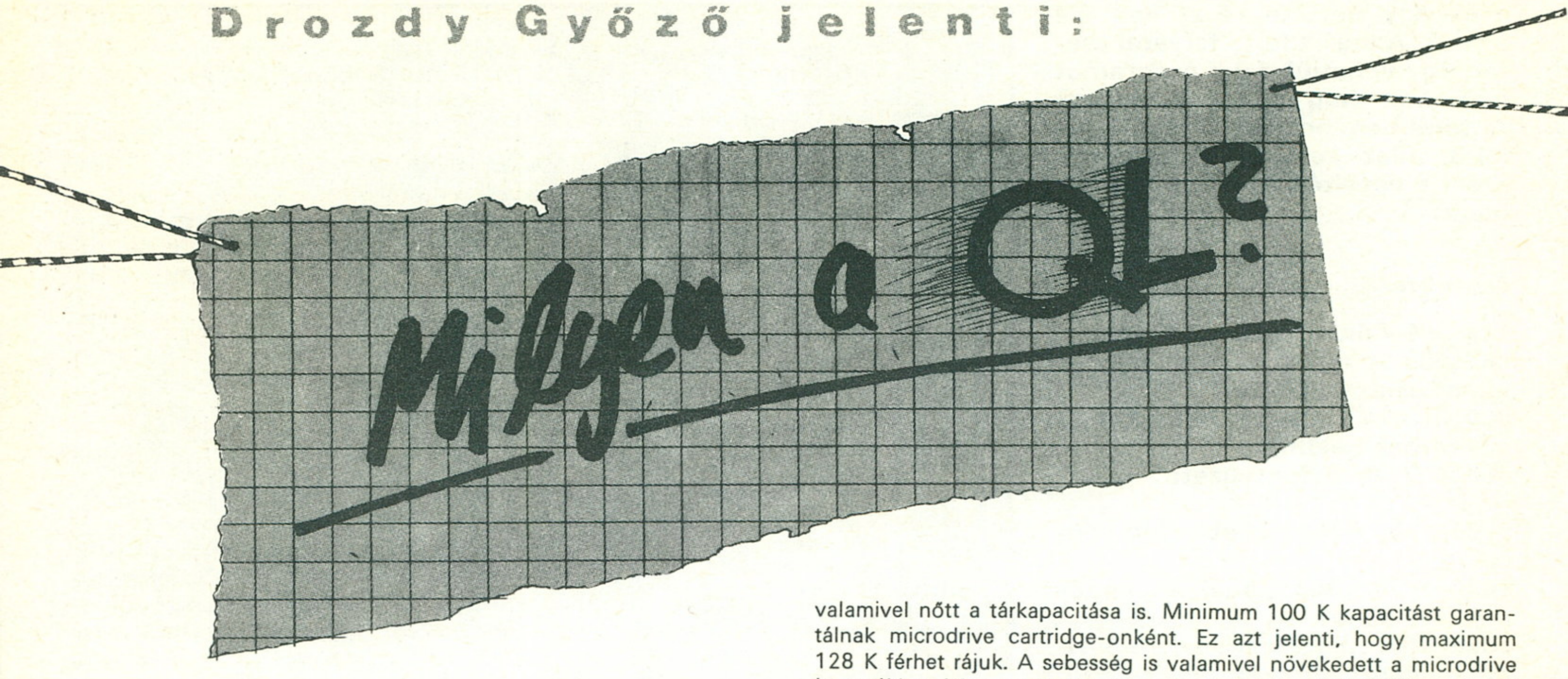
NX	X irányú nagyítás
NY	Y irányú nagyítás
MC	Multicolor (többszínű)
SZ	A szín kódja
S1, S2	Multicolor színei
YS	Az Y irányú vezérlő karakterek
XS	Az X irányú vezérlő karakterek
TS	A lenyomott gomb
A	Annak a karakternek a képernyő kódja, ahol a kurzor áll
V	A kör (SHIFT Q) ASCII kódja
X Y	A kurzor koordinátái
P	Azt adja meg, hogy a pont vízszintesen a sprite hanyadik harmadában áll.
ME	A hely, ahova tölteni kell a sprite-ot
ER	Az érték, amit tölteni kell
A ( )	A sprite 63 adata

Kellemes manótervezést, és sok sikert!

***Szabó Gál András,* Eötvös József Gimnázium II. D.**

```
1110 A=PEEK(Y*40+X+1024):IF A=81THENV=209
1120 PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X);"■ ";"■";CHR$(V )
1125 IF PEEK((Y+1)*40+X+1024)=209THEN 1135
1130 Y=Y+1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)".";:
1133 RETURN
1135 Y=Y+1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)"●";
1140 RETURN
1200 V=160
1210 A=PEEK(Y*40+X+1024):IF A=81THENV=209
1220 PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X);"■ ";"■";CHR$(V )
1225 IF PEEK(Y*40+X-1+1024)=209THEN 1235
1230 X=X-1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)".";:
1233 RETURN
1235 X=X-1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)"●";
1240 RETURN
1300 V=160
1310 A=PEEK(Y*40+X+1024):IF A=81THENV=209
1320 PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X);"■ ";"■";CHR$(V )
1325 IF PEEK(Y*40+X+1+1024)=209THEN 1335
1330 X=X+1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)".";:
1333 RETURN
1335 X=X+1:PRINT"■"LEFT$(Y$,Y);LEFT$(X$,X)"●";
1340 RETURN
1400 PRINT"■"LEFT$(Y$,Y)LEFT$(X$,X)"●";
1410 X1=X-1:Y1=Y-1
1420 IFX1/8<=3THENP=2
1423 IFX1/8<=2THENP=1
1425 IFX1/8<=1THENP=0
1430 HE=(P+Y1*3)+832
1440 E=(8-(X1-(P*8))) :ER=PEEK(HE)OR2↑E
1450 IF HE>=832ANDHE<960THENPOKEHE,ER
1470 RETURN
1500 PRINT"■"LEFT$(Y$,Y)LEFT$(X$,X)"■ ";
1510 X1=X-1:Y1=Y-1
1520 IFX1/8<=3THENP=2
1523 IFX1/8<=2THENP=1
1525 IFX1/8<=1THENP=0
1530 HE=(P+Y1*3)+832
1540 E=(8-(X1-(P*8))) :ER=PEEK(HE)AND(255-2↑E)
1550 IF HE>=832ANDHE<960THENPOKEHE,ER
1570 RETURN
1600 PRINT"■"LEFT$(Y$,Y)LEFT$(X$,X)"■ ■";
1610 RETURN
7000 FORI=0TO24:PRINTSPC(40):NEXT
7010 PRINT"■╭──────────────■FELVEVO RUTIN■────────────────╮";
7020 PRINT"|██████████████████████████████████████|";
7030 PRINT"|██████████████████████████████████████|";
7032 PRINT"|██████████████████████████████████████|";
7033 PRINT"|██████████████████████████████████████|";
7034 PRINT"|██████████████████████████████████████|";
7035 PRINT"|██████████████████████████████████████|";
7040 PRINT"|            ╭──────────╮              |";
7050 PRINT"|            | ■P■ROGRAM |               |";
7060 PRINT"|            ╰──────────╯              |";
7070 PRINT"|              VAGY    FILE MODBAN AKAROD|";
7080 PRINT"|            ╭──────────╮              |";
7090 PRINT"|            | ■A■DAT    |               |";
7100 PRINT"|            ╰──────────╯              |";
7110 PRINT"|██████████████████████████████████████|";
7120 PRINT"|██████████████████████████████████████|";
7130 PRINT"|■■■AZ ■INVERZBEN■ ALLO BETUT NYOMD LE!  |";
7140 PRINT"|██████████████████████████████████████|";
7150 PRINT"|██████████████████████████████████████|";
7160 PRINT"|██████████████████████████████████████|";
7170 PRINT"|██████████████████████████████████████|";
7180 PRINT"|██████████████████████████████████████|";
7185 PRINT"|██████████████████████████████████████|";
7190 PRINT"╰───■(C)COPYRIGHT CITY SOFTWARE 1985■─────╯";
7200 GETT$:IFT$=""THEN7200
7210 IFT$="P"THEN 8000
7220 IFT$="A"THEN 8100
7230 GOTO 7200
8000 INPUT"■■■HANYAS SORSZAMTOL IRJAM KI";S
8010 FORI=0TO62STEP3::A=832
8020 PRINTI+S;"DATA"PEEK(A+I)","PEEK(A+I+1)","PEEK(A+I+2)
8030 NEXT:NEW
8100 FORI=0TO62:A(I)=PEEK(832+I):NEXT
8110 OPEN1,1,1,"SPRITE FILE"
8120 FORX=0TO62
8130 PRINT#1,A(X)
8140 NEXT
8150 CLOSE 1
```

Drozdy Győző jelenti:



A Sinclair QL gép egyszer még Vallatónkban is fog szerepelni. Ma még azonban olyan kevés példány van belőle az országban, hogy vallatásra nem mertünk gondolni. Ehelyett megkértük egy régi ismerősünket, aki néhány hónapja ismerkedik egy ilyen géppel, hogy írja le mindazt, amit tud róla, s amit tapasztalt.

1984 februárjában jelentette be a Sinclair cég, hogy hamarosan piacra hozza legújabb személyi számítógépét a Quantum Leapet, 400 fontos áron. Megrendeléseket is elfogadott rá. A QL azonban, némi előzmények után, igen nehezen került csak piacra. Állítólag a szoftverrel voltak problémák. Később ki is derült, hogy az eredetileg fenntartott 32 K ROM-ba nem fér bele a szoftver, és ezért ezt a későbbiek során 48 K-ra bővítették. A gépnek volt több változata, amelyek nem voltak teljes értékűek. A végleges változat múlt év októberében jelent meg a londoni nagy komputerkiállításon. A Sinclair cég eredetileg havi 100 ezer darabos gyártásra akarta felfuttatni a termelést, de tavaly októbertől idén februárig mindössze 50 ezer darabot adott el. Ez valószínűleg annak köszönhető, hogy a gép bevezetése csapnivalóan rossz volt, rengeteget kellett várni rá, hitegették az embereket, és rossz verziókról készült tesztek jelentek meg a különböző komputerújságokban. A kudarc alapján idén februárban a Sinclair cég úgy döntött, hogy újra indítja a QL-t teljesen új reklámstratégiával. Ennek eredménye az lett, hogy a QL-tulajdonosok automatikusan a QLUB klub tagjai lesznek – ezért korábban 35 fontot kellett fizetni –, mostantól kezdve a gép megvételével ez ingyenesen bekövetkezik.

Adatok

Röviden tekintsük át a gép műszaki paramétereit. A központi egység a Motorola 68008-as mikroprocesszora. Erről a mikroprocesszorról a Sinclair azt állítja, hogy 32 bites. Ez csak többé-kevésbé felel meg a valóságnak. A belső architektúrája tényleg 32 bites, viszont az adatvonalai csak 8, a címvonalai pedig 20 bitesek. Lényegében ezt a processzort 8 és 32 bit között bármilyennek lehet nevezni, mindegyikben van több-kevesebb igazság. A gépben van egy második processzor is, ami az Intel 8049-es processzora, ami a keyboard kontrollt végzi, hangot generál, és az RS–232C bemeneteken vevőként működik. 128 K RAM van beépítve, ami 640 K-ra bővíthető. A ROM 48 K-s, amiben megtalálható a szuper BASIC interpreter és a QDOS-nak nevezett operációs rendszer. A Szuper BASIC a BASIC-nek egy jelentős továbbfejlesztése, megszabadul a BASIC azoktól a jellemzőitől, ami miatt csak kezdők számára ajánlható programozási nyelv volt, például eljáráshívásokra megfelelő paraméterátadással lehetőség van. A QDOS operációs rendszer egyfelhasználójú, de az egy felhasználó több feladatot is futtathat egyszerre, ennek támogatására különböző ablakokat tud kezelni az operációs rendszer a képernyőn. A gépbe beépítettek két darab microdrive-ot, amely a Spectrumnál megismert microdrive továbbfejlesztett változata, valamit gyorsítottak rajta, és valamivel nőtt a tárkapacitása is. Minimum 100 K kapacitást garantálnak microdrive cartridge-onként. Ez azt jelenti, hogy maximum 128 K férhet rájuk. A sebesség is valamivel növekedett a microdrive kazettákhoz képest: 15 kbyte/sec. A gépnek egy QWERTY rendszerű billentyűzete van, amely 85 gombot tartalmaz, amiből 5 úgynevezett funkciógomb és 4 kurzorgomb. A gombok teljes mértékben mozognak, a keyboard professzionálisnak látszik, kényelmes rajta a gépelés. Tehát a Spectrumhoz képest itt egy lényeges előrelépés történt. A gépnek két soros interface-e (RS–232C) van. Ezenkívül rendelkezik microdrive hozzákötésére alkalmas porttal, ahová még további 6 microdrive csatolható. 1 ROM dugaszolható a hátuljába, ahol felhasználói szoftvereket helyezhetünk el 16 kbyte terjedelemben, ezenkívül Spectrumhoz, illetve más QL gépekhez lehet csatlakozni a hálózati csatlakozón keresztül. Két darab joystick is hozzákapcsolható. Kétféle üzemmódja van a QL-nek, egyikben normál háztartási televíziót használunk, másikban RGP jelet és monitort. A lényegesen különböző felbontásokhoz természetesen igazodik a gép. A monitoron 25 sor és 85 oszlop jelenik meg, míg a tévéképernyőnél kevesebb oszlop (változtathatóan 40 és 60 között). Ennek következtében még tévéképernyőn is jól olvashatóak a feliratok, a monitoron viszont impozánsan sok információ fér egyszerre fel a képernyőre. A kép felbontása 512x256, ha csak négy színt akarunk használni, és 256x256 8 szín esetén. A felbontások átkapcsolhatók. Minden egyes géphez a gyártó cég mellékel négy darab felhasználói szoftvert. Ezek a QL Quill, amely egy szövegszerkesztő program, a QL Abacus, amely egy könyvelői táblázatkezelő program, a QL Easel, amely egy business grafikakészítő és a QL Archive, amely egy file-kezelő program. Ez a négy felhasználói szoftver jelentős értéket képvisel, mert nagyon jól megírtnak tűnik, és széles körben felhasználható. Például a szövegszerkesztő tud aláhúzott betűket, vastag betűket, alsó és felső indexet írni. Ezzel, hogy ha megfelelő nyomtatót is kapcsolunk a géphez, például az FX 100-ast, teljesen publikációkész formát kaphatunk, nincs szükség később az indexeknek a beírására. Sajnos a Quill magyar karakterek beírására alkalmatlan, sőt az angol karakterkészletből is csak a fontosabb jeleket ismeri, még azoknak a karaktereknek egy részét sem lehet vele bevinni, amiket egyébként a QL ismer. Nyilvánvalóan megoldható ennek a szövegszerkesztőnek az átdolgozása, hogy a magyar karaktereket is hajlandó legyen elfogadni. (Ha valaki ezt már megtette, legyen szíves, tájékoztasson róla.) A négy alkalmazói szoftverről megjelent tesztek mind elismerően nyilatkoztak. Egy kifogás merült fel velük szemben, hogy a betöltésük nagyon hosszú időt vesz igénybe. Tényleg, körülbelül egy percig is eltart, amíg elindulnak. A QL újraindításának egyik eredménye az lett, hogy megjelentek a 2-es verziószámmal jelölt alapszoftverek, amelyek az eredeti 1-estől annyiban különböznek, hogy teljes mértékben gépi kódban íródtak. Így betöltésük ideje jelentősen lerövidült. Futásuk még gyorsabb lett, habár az eredetieknél sem tudtam a sebességüket kihasználni. A klubtagok számára a 2-es verziószámú szoftverek ingyen elérhetők. Aki nem klubtag, annak 50 fontba kerül. Én a módosított változatokat még nem ismerem, ha van az olvasók között valaki, akinek ezek már megvannak, örülnék, ha megkeresne.

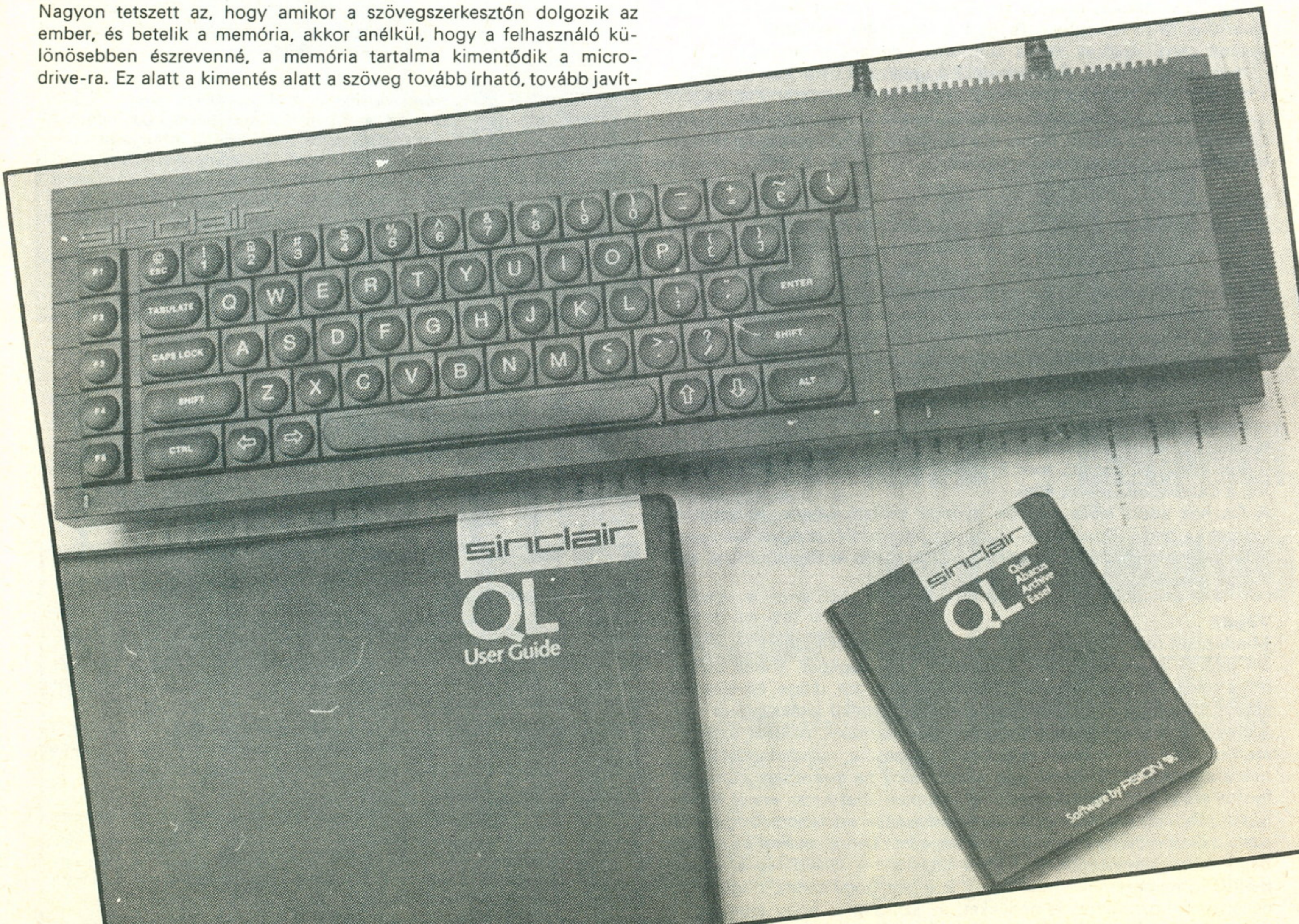
Benyomásaim

Nézzük, a műszaki adatok után milyen benyomásaim voltak a QL-lel kapcsolatban. Bekapcsolás után a QL sokkal csupaszabbnak tűnik, mint a Spectrum. A Spectrumhoz mellékeltek egy demókazettát, ami már nagyon sok mindennel megismerteti a felhasználót, és egyben elképzelést nyújt arról, hogy mit lehet a géppel produkálni. Sajnos ilyen a QL-hez nincs, ezért a vele való megismerkedés és az első benyomások gyűjtése is sokkal lassúbb volt. Továbbá a QL komoly felhasználásokra alkalmas igazán, amikhez nagyobb előkészületek szükségesek, mint a játékokhoz. Mindezt azért bocsátom előre, mert kéthónapi ismeretség után még nem merném azt mondani, hogy ismerem a gépet, csak most kezdem megismerni. Az első benyomásom, hogy megfelelően körbeépítve, azaz monitort, megfelelő mátrixnyomtatót csatlakoztatva hozzá, és esetleg a memóriát kibővítve egy olyan konfiguráció alakulhat ki, ahol már nem lehet vitás az, hogy milyen komoly dologra lehet használni. A BIT-LET hasábjain nemrég vita folyt arról, hogy milyen komoly felhasználása lehet a személyi számítógépeknek. Meggyőződésem az, hogy ez a kiépítéstől függ. Egy QL az előbb vázolt kiépítés mellett már igenis nagyon komoly dolgokra használható, pl. a levelezés ezzel folytatható, tudományos publikációk ezen készíthetők, anyag-, könyvtárnyilvántartások készíthetők rajta, nagyon eredményes módon. A szuper BASIC nyelv, a multitasking lehetőség és a két processzor olyan felhasználási lehetőségeket is megnyit, amelyeket más személyi számítógépekkel nem nagyon lehet elképzelni, illetve csak ennél lényegesen drágább gépekkel voltak eddig elképzelhetők.

Például a szuper BASIC egyik kedvező tulajdonsága, hogy gépi kódú részletekkel kiegészítve továbbfejleszthető, olyan gépi kódú részek csatolhatók még hozzá, amelyek a továbbiakban a szuper BASIC alap-utasításkészletével egyenrangú utasítások. Így egy nagyon rugalmas rendszert tudunk létrehozni a szuper BASIC-ben.

Nagyon tetszett az, hogy amikor a szövegszerkesztőn dolgozik az ember, és betelik a memória, akkor anélkül, hogy a felhasználó különösebben észrevenné, a memória tartalma kimentődik a microdrive-ra. Ez alatt a kimentés alatt a szöveg tovább írható, tovább javítható és a felhasználó tulajdonképpen csak a microdrive hangjáról és lámpájáról veszi észre, hogy most a szövegszerkesztő a microdrive-hoz fordult. Közben éppen úgy lehet használni, mintha mi sem történt volna. Ezzel jelentős időt lehet megtakarítani. Ugyanígy programok ki- és bevitelénél a microdrive-ra, a parancs kiadása után a rendszer további parancsok elfogadására és végrehajtására képes, ami a microdrive-ra való várakozási időt még lerövidíti. A microdrive az igazi diszkek tárolókapacitásától és sebességétől messze elmarad, de összehasonlítva például a Commodore-ral, a Commodore discjéhez képest mintegy tízszer gyorsabb. De eddig az volt a baj, hogy azonos tárolási kapacitás kb. tízszer annyiba került, mert egy microdrive cartridge 5 font volt. Most az újraindítás óta a microdrive cartridge ára 2 font, ami végleg eldöntötte a Sinclair microdrive és a Commodore disc közötti versenyt.

A következő érdekes téma a gép sebessége. 7,5 MHz-cel működik a CPU és miután ez 32 bites architektúrájú, jelentős sebességnövekedésre számíthat a felhasználó. Lefuttattam a BIT-LET-ben korábban megjelent Benchmark programokat és általánosságban azt lehet mondani, hogy a Spectrumhoz képest a futásidők – csalódást keltve – csak kevesebb mint felére csökkentek, így nagyjából a korábbi jobb gépek idejének környékére esnek. Kivéve a számításigényes 8. programot, ahol kevesebb mint tizedére esett a futásidő a Spectrumhoz képest. Ez nyilvánvalóan a 32 bites processzornak köszönhető, hiszen a számítási műveleteknél főleg a processzor belső architektúrája számít. A futási idő csak ilyen kismértékű javulása kicsit csalódást jelent első látásra, de meg kell jegyezni azt, hogy a szuper BASIC-ről azt állítják, hogy egy programrészlet futásideje független a program hosszától, míg a korábbi BASIC-variációknál a program hosszának növekedésével jelentősen nőtt az ugyanilyen programrészletek



Drozdy Győző jelenti:

Milyen a QL?

futtatásához szükséges idő. Mivel általában a futásidő akkor kritikus, ha hosszúak a programok, ezért a valóságban a sebességnövekedés a Benchmark teszteknél tapasztaltnál lényegesen nagyobb lehet. További sebességnövelő tényező a multitasking lehetőség, amit ha csak olyan egyszerű esetekben használunk ki, mikor a microdrive-ra viszünk ki vagy hozunk be valamit, és eközben számolunk valami mást, akkor is döntő javulást kaphatunk.
A QL grafikája feltűnően gyors. A Spectrumhoz képest lenyűgöző sebességet produkál. Úgy tűnik, hogy szuper BASIC-ben programozva olyan gyors, mint a Spectrum grafikája gépi kódban programozva, pedig azt mindenki látta, hogy micsoda jól mozgó játékok készíthetők Spectrumra gépi kódú programozással.
A gép felhasználását nagymértékben befolyásolja az, hogy milyen kiegészítő berendezések, egységek kaphatók hozzá, illetve milyen szoftverek érhetők el. Mindenekelőtt megjelentek a memóriakiterjesztő kártyák, az 512 K-val való kiterjesztés – amivel a gép eléri a maximális 640 K RAM-ját –, az általam látott legolcsóbb ár szerint 365 font, így tulajdonképpen 765 fontért egy több mint fél Mbyte RAM-mal rendelkező géphez juthatunk, ami nagyon kecsegtető. A géphez kapható floppy disc illesztő egység is, amely képes a különböző méretű floppy disc drive-okat is illeszteni, és természetesen drive-ok is vásárolhatók hozzá. Tehát végső soron floppy disc periféria is kiépíthető a microdrive-tárolás mellett. Ezenkívül 1200 fontért Winchester disc illesztő és drive is kapható és így 7,5 megabyte-os háttértárolóra tehetünk szert. Árulnak hozzá centronix interface-t, ami egy szoros, párhuzamos átalakító, ezen keresztül nagyon sok nyomtató illeszthető a géphez. IIIE interface is kapható hozzá. Megjelent egy 3 egységből álló interface, amellyel telefonvonalon keresztül más számítógépekhez tudjuk kapcsolni a QL-t, illetve egy QL-ekből álló számítógéprendszer kiépítése van folyamatban Angliában.

Szoftver

Nézzük ezek után a szoftverkínálatot. Többféle assembler is kapható a géphez, egy teszt szerint a Metacomco cég assemblere a legjobb. Magas szintű nyelvek közül a következők érhetők el hozzá: BCPL, LISP, PASCAL-ból több verzió is kapható, FORTH, APL és a C nyelv. Hirdetik, hogy hamarosan meg fog jelenni – Angliában valószínűleg meg is jelent – a FORTRAN. Tekintettel arra, hogy a QL nem olyan régen érhető el ténylegesen, megnyugtató dolog, hogy már ilyen sok magasszintű nyelv kapható hozzá. Ez valószínűleg annak a következménye, hogy a 68 ezres Motorola család teljesen szoftver-kompatibilis, és így más rendszereken kifejlesztett 68 ezres szoftverek könnyen áttelepíthetők QL-re. Azt hiszem, ez nagyon fontos érv a QL mellett, mert így jelentős és komoly szoftverkészlet fog a QL rendelkezésére állni.
A QL-hez adott leírás nagyon gyönge kivitelű sajnos, lényegesen rosszabb, mint a Spectrumé. Bizonyos információkat egyáltalán nem lehet benne megtalálni. Remélem, előbb-utóbb ki fognak hozni egy sokkal jobbat is.

Végül

Mindezeket összefoglalva a QL-ről a következőket tudnám mondani: aki játékra akarja használni a személyi számítógépet, annak biztos, hogy nem éri meg QL-t venni, mert ahhoz túl drága és olcsóbb számítógépekhez lényegesen szélesebb választékú játékszoftver van. Aki viszont valami komoly célra akarja használni, például szövegszerkesztésre, tudományos számítások végzésére, nyilvántartásra, annak a számára a QL ideális gép akkor, hogy ha a rendszer olcsó ára fontos szempont. Természetesen drágábban lehet kapni ennél lényegesen jobb rendszereket is. A QL fő erényének azt tartom, hogy a 68 ezres mikroprocesszor család egyik tagja van benne, ezáltal a szoftver terén óriási lehetőségek vannak a QL-ben. A QDOS és a szuper BASIC jó rendszernek tűnik. Az az érzésem, hogy ahhoz, hogy a QL beváltsa a hozzáfűzött reményeket, az szükséges, hogy legalább 1 millió darabot eladjanak belőle, mert akkor lesznek igazán olcsók a hozzá való szoftver- és hardvertermékek. Mi, QL-felhasználók mindenesetre bízunk ebben.

NYÍLT TÉR

CHIP-kelődés

Huszonkilenc éves matematika–technika szakos tanár vagyok egy békéscsabai iskolában. Az 1984. decemberi BIT-LET vezércikke felborzolta az idegeimet, és nagyon megkeseredett a szám íze. Húsz srácom van, akik a szakkörben már megszállottként szerelgetik a kis alapáramköreiket. Csodálatos dolog számukra, amikor a saját paneljukat maratják, még remeg a kezükben a páka, mert a forrasztás borzasztóan izgalmas dolog. Jó látni őket, amikor a deszkapanel beindult, villog, fütyül stb. stb. ...

De én vagyok a bűnös, mert megfertőztem őket, és arra ítéltetem ezeket a gyerekeket, hogy előbb-utóbb a zsebpénzüket alkatrészekre költsék. Költenék! Akárcsak jómagam, már 15 éve költeném! És ráadásul jön egy illetékes szakember (vezető!) és le akarja húzni, húzatni a rolót. Pedig ezt a rolót eddig még fel sem húzták.

Szerencsétlenségemre csatlakoztam a Homelab Kit építőkhöz! Mivel nem dolgozom mikroelektronikai vállalatnál, „arra vagyok ítélve", hogy boltból szerezzem be az alkatrészeket. Csinálnám én a saját anyagi erőmből (feleségem egyetértésével!) is, időt, fáradságot nem sajnálva, csak LENNE MIBŐL! Jól felszerelt iskolában dolgozom, az iskola kinyögött a költségvetésből két darab számítógépet (Sinclair). Erre jut 34 gyerek, mind kis számítógépbarát szakköri tag, s ezen kívül még 14 gyerek, aki hivatalos számítástechnika fakultációban tanul. Ne higgyék, hogy telhetetlen vagyok! Tudom, vannak gyengén felszerelt vidéki iskolák! De az én iskolámba 1200 gyerek jár és 2 év múlva már több mint 1600 gyerek! Minimum 10 százaléka igen tehetséges tanuló, és meg kell adni nekik a lehetőséget a fejlődésre.

Magam nem tudok több gyerekkel foglalkozni, mert kevés a gép. Így is tülekednek, hogy odaférjenek. Van, akit a programozás bolondított meg, van akit a komputer szerkezete. Ők felsőbb iskoláikban már biztos szeretnének építeni! Mind a két gyerektípus nagyon fontos a jövőnk szempontjából. Számomra egyszerűen hihetetlen és érthetetlen, hogy amikor a fejlett országok műszaki színvonala hatványozott sebességgel hagy el bennünket, akadnak olyan középvezetői iparpolitikai szakemberek, akik pusztán még több korlát felállításában látják a megoldást!
Maradok továbbra is hű olvasójuk

***Végh Sándor,** 5600 Békéscsaba, Paróczay u. 7.*

PROGRAM CSERE-BERE

Tisztelt Olvasótársak!

Sinclair Spectrum számítógépemhez van két programom (OMNICALC; PAINTBOX), amit sajnos nem tudok használni, nem sikerült eddig megszereznii róluk a leírást. Ha valaki rendelkezne e két program leírásával, kérem, segítsen, szeretnék egy fénymásolatot készíteni róla.

***Kiss István Tibor,** Solymár, Vasút utca 62/a. 2083*

Commodore VC–20 és C–64-es programokat cserélek szalagon.

Molnár Tibor Vésztő, Eötvös u. 4/A. 5530

Folyóiratokban megjelent Apple II, TRS 80, TI 99, Dragon 32, Atari, Spectrum, ZX81, Sharp HZ 700, Sharp PC 500, VIC 20, C 64 programokat adnék C 64 programokért. Valamint C 64-es kollégák ismeretségét keresem.

***Majtényi György,** Kisújszállás, Szabadság téri lakótelep 8. 5310*

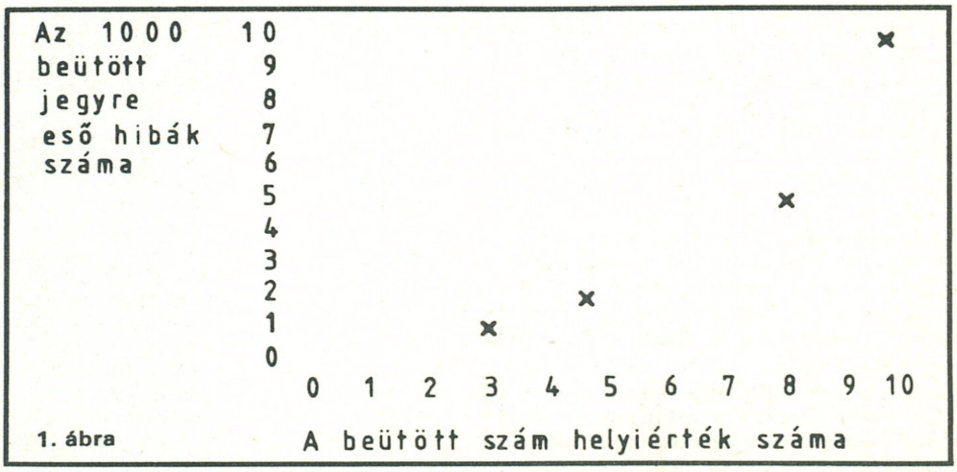
VONALKÓD-ALKALMAZÁS

MIÉRT PONT VONALKÓD?

Rendszereinkben a vonalkód alkalmazásának célja az egyértelmű és egyszerű *azonosítás.* Mint közismert: abban az esetben, ha egy számítógépbe hibás adat kerül be, akkor a kapott eredmény is hibás. A szakirodalom ezt tömören a GIGO (Gerbage In–Gerbage Out: Szemét Be–Szemét Ki) elvben fogalmazza meg.

Billentyűzetről bevitt számok esetén fellépő hibák eloszlását az 1. ábra szemlélteti, amely az NSZK-beli METO Gmbh. vizsgálatainak eredményei alapján készült. Ebből jól látható, hogy a beütött számok helyiérték számának növekedésével igen megnövekszik a hibák száma. A vonalkód alkalmazásával viszont a hibás adatbeadások száma egy adott rendszerben nagymértékben csökkenthető a billentyűzeten keresztüli adatbevitelhez képest. Sok rendszerben a vonalkód választását – egyéb gépi azonosítással szemben – egyszerűsége, megbízhatósága, tömörsége és így a gazdaságos gépi leolvasás lehetősége indokolja. A kód az információt (adatot) egy dimenzióban, irányfüggetlenül tartalmazza. Önszinkronizáló jellege pedig gyors, és nem túl nagy precizitást követelő leolvasást tesz lehetővé.

Ennek az új technikának a bevezetésekor tehát a kezelők nem érzik terhesnek e rendszer kezelését, így a számítógép számukra is érzékelhető előnnyel jár. Számos területen a vonalkód alkalmazása az egyetlen gazdaságos út a számítástechnika előnyeinek kihasználásához.



1. ábra

FELHASZNÁLÁSI LEHETŐSÉGEK

A vonalkód-alkalmazás néhány tipikus területe

- cikkszám szerinti árunyilvántartó rendszer
- gyártásellenőrzési és -irányítási rendszer
- leltári rendszer (állandó leltár)
- rendezés és vezérlés (konvejor, szállítólánc)
- hozzárendelési feladatok (kórház)
- jogosultságellenőrzés
- ki-, ill. beviteli nyilvántartás (könyvtár, szállítás)
- billentyűzetkiváltás speciális okokból. (Pl. erősen rázkódó, a billentyűzet számára elviselhetetlen környezet; a billentyűzet számára nem elegendő a hely stb.)

A felsorolt lehetőségek csak néhányak a sok lehetséges közül. Természetesen a vonalkódok minden olyan területen is alkalmazhatók, ahol viszonylag korlátozott mennyiségű információt kell megbízhatóan, gyorsan, olcsón, mobil módon bevinni. Az alkalmazhatóság szempontjából fontos kérdés a jelölési rendszer (vonalkód és a vonalkódban megtestesülő információ) teljessége. Egy kis, *zárt* rendszerben – ahol a rendszer megvalósításához tartozó minden tevékenység lokális érdeket szolgál – nem jelentkezik nehézség. A nagy, *nyitott* rendszerekben – mint például a kereskedelem, a szolgáltatás stb. – az országos és a nemzetközi szabványok az iránymutatók. Ilyen szabvány például az EAN (European Article Numbering) vonalkód szabvány, amely Magyarországon 1985-ben jelenik meg. Magyarország már korábban csatlakozott az EAN szervezethez. Az EAN vonalkóddal lehetséges az ún. Egységes Termékazonosító Kód (ETK) megjelenítése is. Erre vonatkozóan az Ipari Közlönyben a következő közlemény jelent meg: *„9045/1984. (IpK. 19.) IpM 1. A kormány Gazdasági Bizottságától kapott megbízás alapján a gazdaságirányításért felelős tárcák irányításával 1986 végéig a termelés és forgalmazás meghatározott körében ki kell dolgozni és be kell vezetni az országosan egységes termékazonosító kódrendszert. (A továbbiakban: ETK) Az egységes termékszámok képzési és használati mechanizmusáról, az 1984–1986-ban bevonásra kerülő termékkörökről az Országos Anyag- és Árhivatal 8001/1984. (ÁT 25.) ÁH számú tájékoztatója ad eligazítást."*

Mindezekből jól látható, hogy a vonalkód egy olyan adathordozó, amely az ipar és a szolgáltatások (kereskedelem) összes területén előbb-utóbb helyet követel magának.

2. ábra

4 012345 678901

VONALKÓD TÍPUSOK

A világban sokféle kód ismeretes. Ezek közül a legelterjedtebbek a következők:

- A köznapi életben leggyakrabban az EAN (l. a 2. ábrát) és az azzal konform UPC kóddal találkozunk.
- Zárt rendszerekben elterjedt típusok az „5-ből 2" vonalkód különböző formái (l. a 3. ábrát). Ennek az INTERLEAVED változata a nemzetközi szállításban is egyre nagyobb szerepet kap.
- Fototechnikai és orvosi alkalmazásokban a CODABAR vonalkód a legelterjedtebb.
- A „39-es kód" (l. a 4. ábrát) a betűk kódolását is megoldja.

3. ábra

a) b) c) d)

4. ábra

Természetesen a kódokat nemcsak a kódolás módja szerint különböztethetjük meg, hanem a következők szerint is:

- A kód tömörsége
- A kód mérete
- A kódolható jegyek száma
- A nyomtathatóság
- Az olvashatóság
- A hordozó anyag

M08X

Felvilágosítást ad:

Sci–L
Vevőszolgálat
1011 Budapest
Iskola utca 10.
Telefonszám: 260-000
Telexszám: 22-4590

A VONALKÓDNYOMTATÁS ESZKÖZEI

A vonalkódnyomtatást az SZKI a rendszereiben – igénytől függően – kétféle módon oldotta meg:

- Hagyományos számítógép-nyomtatóval
- Speciális vonalkód-nyomtatóval

A nyomtató programok az SZKI fejlesztésű MP–80 mátrixnyomtatóhoz, ill. a japán gyártmányú C ITOH 8510 nyomtatóhoz készültek és az M08X, a PROPER 8 (8 bites), valamint a PROPER 16 (16 bites) professzionális személyi számítógépen futnak. A programok az EAN kódolás és az „5-ből 2" kódolás elvének megfelelő kódok nyomtatására alkalmasak.

Ezek a programok egyrészt mint interaktívan futó önálló programok, másrészt mint alkalmazói programhoz szerkeszthető modulok működnek.

Nagy mennyiségű vonalkódot hordozó etikett előállítására speciális vonalkód-előállító berendezések csatolhatók az SZKI professzionális személyi számítógépeihez.

VONALKÓDOLVASÁS ESZKÖZEI

Több éves tapasztalat alapján fejlesztett eszközeink a vonalkódolvasó ceruzára alapozottak: leolvasás egy ceruzaszerű kézi leolvasóval történik.

A vonalkódolvasó az olcsó SZKI termékhez, a TELETERM terminálhoz csatolható, ahol az olvasó funkcionálisan a billentyűzettel VAGY kapcsolatban van.

A vonalkód használatának terjedése fokozatos, sokszor az igény vagy a lehetőség csak a rendszer üzemszerű használata során születik meg. Ennek felismerése vezette az SZKI-t az M08X és a PROPER 8 professzionális személyi számítógépekhez közvetlenül csatlakoztatható vonalkód-berendezés kifejlesztéséhez. Az installálás nem igényel sem hardver, sem szoftver rendszertechnikai átalakítást. Ha valami vonalkóddal kódolható egy adott, már meglevő rendszerben, akkor ezzel a billentyűzettel VAGY kapcsolatban levő berendezéssel a kódolt információ beolvasható a számítógépbe (l. az 5. ábrát).

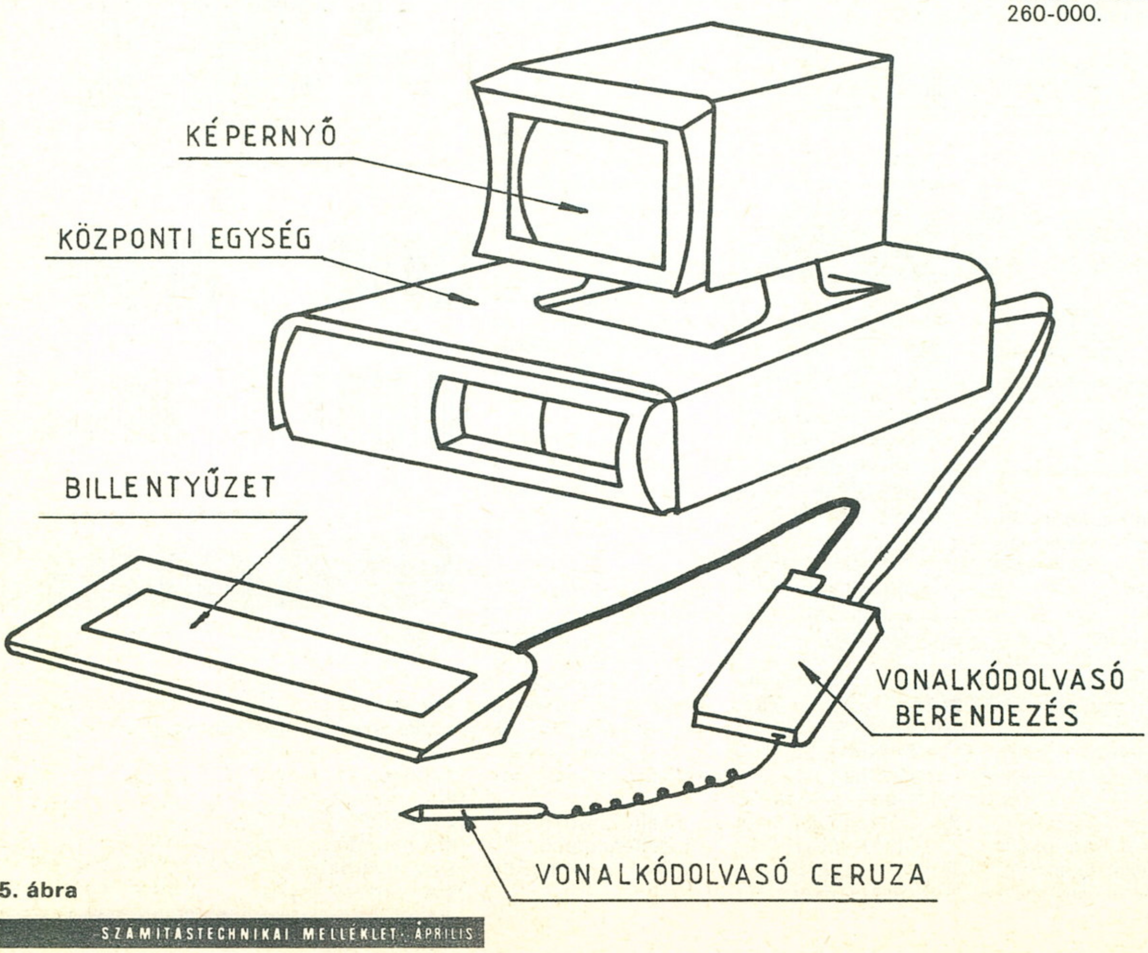
A berendezés csatlakoztatható a számítógéphez párhuzamos interface-en keresztül is, így lehetőség van más rendszertechnikai megoldásokra is.

A berendezéseink az EAN, az „5-ből 2" és a CODABAR kódok értelmezésére készültek fel, de természetesen más kódok használata is lehetséges.

HARDVER, ILL. SZOFTVER KÖRNYEZET

A programok az M08X, a PROPER 8, a PROPER 16 személyi számítógépeken PROPOS 8, ill. PROPOS 16 operációs rendszer alatt futnak.

Felvilágosítást ad: Számítástechnikai Koordinációs Intézet Hardver Laboratórium. Tel.: 260-000.



5. ábra

POSTA

Rendelkezem ZX81-re gépi kódú programozáshoz monitor-programmal. Ez sok ZX-esnél fut. Ezzel kapcsolatban felmerült bennem, hogy érdemes lenne EPROM-ba égetni (természetesen a belső ugrások átírásával), és a 16 K-s bővítőhöz hasonlóan felcsatlakoztatni a gépre.

Véleményem szerint hardverszempontból nem komoly feladat. Gondolom, ennek többen vásárlói lennének. Hasonlóan el tudnám ugyanezt képzelni a nagy felbontású grafikával.

Ezen felül sok olyan programom van, amelyet dokumentumok híján csak részben tudok használni. Szívesen venném, ha a lap hasábjain ezekről is olvashatnánk.

Bakos Sándor, Makó, Kálvin út 36. 6900

Sok más olvasónktól is kapunk hasonló javaslatokat. Használati utasítások ügyében azt tudjuk vállalni, hogy a leveleket, amelyekben ilyet keresnek vagy ajánlanak – névvel, címmel – közzétesszük. A csatlakoztatható bővítéssel kapcsolatban várjuk vállalkozó kedvű olvasóink leveleit!

HIBAIGAZÍTÁS

Februári számunkban sajnálatos módon a Futkározás című program leírásába súlyos hibák kerültek. Olyan sorszámokra hivatkoztunk a leírásban, amelyek nem is léteztek. Elnézést kérünk a szerzőtől, Gajdos Lászlótól és az olvasóktól is. A helyes szöveg a következő:

A játékban előre definiált karakterek vannak, melyek előállítását a 120–150 sorok végzik.

Az alapkép kirajzolása és a változók értékeinek beállítása a 110 és 190–250 sorokban történik.

A 270 és 490 sorok között van elhelyezve a program fő része, A 280-as sor végzi az irányváltoztatást...

SZOFTVER ÖTLETEK



Tisztelt Olvasóink!

Tapasztalatból tudjuk, hogy mindenki, aki programozással foglalkozik, előbb-utóbb rájön olyan fogásokra, amelyek egy-egy apró részfeladatot oldanak meg, amelyekhez különböző „csalafintaságok" ismerete szükséges. Ugyanakkor nagyon sokan szívesen látnának ilyen megoldásokat, mert amit egyszer valaki kitalált, azt kár még egyszer kidolgozni.

Ezért szeretnénk útjára indítani egy sorozatot (rovatot?), amelynek címe lehetne, mondjuk SZOFTVERÖTLETEK.

Ebben szinte kizárólag Olvasóinkra számítanánk, nem titkoltan elsősorban a középiskolásokra. (Tapasztalatunk szerint ott halmozódott fel a legtöbb ilyen fogás, és ezek cseréje, nyilvánossága nincs igazán jól megoldva.) Itt természetesen nem „bolondbiztos", teljes programokat várunk, hanem kifejezetten a lényegre szorítkozó programsorokat, amelyeket egy nagyobb programba be lehet építeni.

Kérjük tehát, hogy akinek van ilyen a tarsolyában, küldje el címünkre! Ezeket mi összegyűjtjük, és csoportosítva, rendezve közöljük. Csak kipróbált és jól működő programocskákat küldjenek, megfelelő megjegyzésekkel együtt! Ne felejtsék el feltüntetni, hogy milyen típusú géphez, milyen nyelven íródott! Szívesen látunk minden, hozzáférhető programnyelven készült anyagot.

A továbbiakban itt szeretnénk az egyes géptípusoknál előforduló, gépkönyvekben és egyéb irodalomban fellelhető pontatlanságokat, hiányosságokat is közzétenni.

És amivel nem kívánunk foglalkozni: gépkönyvben vagy széles körben terjedő irodalomban közöltek ismételgetése, és programok védelme.

Várjuk tehát Olvasóink SZOFTVERÖTLETEIT!



SZAKKÖRÖKNEK!

INTERRUPT

Mint a bevezetőben már említettük, a Z80 CPU-nak két olyan kivezetése is van, amely működésének megszakítására szolgál. Ezek egyikének működése programból letiltható. A letiltható megszakítást maszkolható interruptnak, a másikat nem maszkolható interruptnak nevezzük.

Ha a CPU nem maszkolható megszakításkérést érzékel, akkor a folyó gépi kódú utasítás befejezése után a PC tartalmát elteszi a stackbe, a PC-be 0066-ot tölt. Ez azt jelenti, hogy egy Call 0066 utasítást hajt végre a megfelelő elektromos impulzus hatására. Pontosabban ilyenkor még az interrupt flip-flop tartalmát is beállítja. A 0066 címen kezdődően kell leírni azt a programrészletet (szubrutint), amit a nem-maszkolható megszakításkérés érzékelésekor a CPU végrehajtani köteles. Ezt a szubrutint a RET utasítás egy módosított változata, a RETN zárja le. Ez az előbb említett flip-flopok helyes kezelését biztosítja. Végrehajtásakor a program futása ott folytatódik, ahol az interrupt kérés észlelésekor tartott.

Ha a CPU maszkolható megszakításkérést kap, ezt csak akkor veszi figyelembe, ha az engedélyezve van. A gép bekapcsolásakor a RESET hatására a maszkolható megszakítás kérés mindig tiltott, annak figyelembevételét mindig programból kell engedélyezni EI (enable interrupt) utasítással. Ezt az engedélyt később bármikor visszavonhatjuk DI (disable interrupt) utasítással. Ha szükséges, akárhányszor ki-bekapcsolhatjuk a megszakításkérés engedélyezését. Azt, hogy engedélyezett interrupt-jel érkezésekor mi történjen, programból befolyásolhatjuk. A CPU ugyanis három különböző interrupt kezelésre képes. Ezeket az IMO, IM1, IM2 utasításokkal választhatjuk ki.

IM0 (interrupt mode 0)

Ennél az üzemmódnál, a megszakítást kérő egység az adatbuszon tetszőleges Z80 utasítást adhat a processzornak.

IM1 (interrupt mode 1)

Ennél az üzemmódnál a megszakítás-kérés egy 0038 H címre vonatkozó szubrutin hívással egyenértékű, a flip-flopok kezelésétől eltekintve. A szubrutin végét RETI jelzi.

IM2 (interrupt mode 2)

Ez az üzemmód teszi lehetővé a legváltozatosabb választevékenységeket. Előkészítésként az I regiszterbe kell töltenünk a kiválasztott szubrutin címtáblázat kezdő címének felső nyolc bitjét. A megszakításkérés elfogadásakor a periféria szolgáltatja a szubrutin címtáblázat címének alsó nyolc bitjét (ez csak páros szám lehet). Az így kialakuló címen kell megadni a szubrutin kezdő címét.

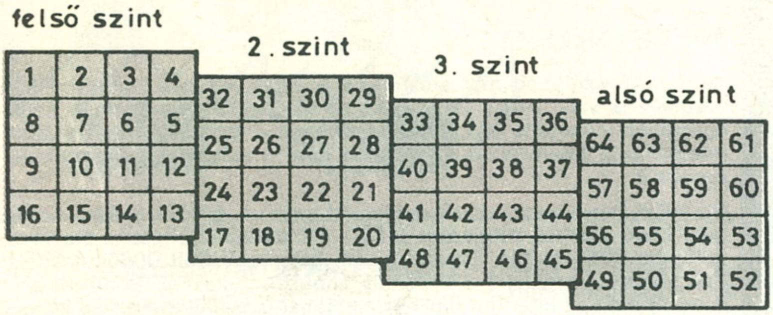
Utasítás	Kódja Tízes	 Hexa
EI	251	FB 4
DI	243	F3H
IM0	237 70	ED 46H
IM1	237 86	ED 56H
IM2	237 94	ED 5EH
RETI	237 77	ED 4DH
RETN	237 69	ED 45H

Ezek az utasítások a flageket nem állítják.
Már korábban említettük, hogy a HALT utasítás hatására a CPU addig vár, amíg interrupt jelet nem kap.

PRIMO-NYERŐ 3. FELADAT MEGOLDÁSA

Állítás: tologatással nem érhetünk el olyan helyzetet, mely az eredetitől csak abban különbözik, hogy két kis kocka meg van cserélődve.

Indoklás:
Számozzuk sorba a nagy kockában a kis kockák helyeit az ábrán látható módon:



Ne felejtsük el, hogy a kis kockák is meg vannak sorszámozva 1-től 63-ig. Ezután „kiterítjük" a játékot egy egyenes mentén, egy játékhelyzetet úgy jellemzünk, hogy sorba leírjuk, hogy az 1-es számú helyen hányas számú kis kocka van, utána, hogy a 2-es számún mi van stb., végül, hogy a 64-es helyen melyik kis kocka van. Ahol nincs kocka, arra az egy helyre nem írunk semmit, ott üres helyet hagyunk. Ezután a játék egy lépése úgy zajlik le, hogy egy számot „megfogunk", és helyéről az üres helyre „tesszük", persze ezt csak akkor tehetjük meg, ha az „átrakandó" szám helye és az üres hely szomszédosak. (Ezt az ábrán gyorsan ellenőrizhetjük.)
A kis kockák egy-egy helyzetéhez hozzárendelünk egy-egy számot, melyet a következőképpen kell kiszámítani: összeszámoljuk, hogy hány olyan kis kockapár van, melyeknél a kisebb sorszámú kocka nagyobb sorszámú helyen van. Ezt a számot nevezzük, mondjuk a helyzet jellemzőjének.

Állítás: bármilyen helyzetben bármelyik (a lyukkal szomszédos helyen lévő) kis kockát toljuk az üres helyre, a helyzet jellemzője páros számmal változik.

Indoklás: az ábrán láthatjuk, hogy páros sorszámú helyek minden szomszédja páratlan sorszámú, s viszont. Így egy kis kocka áttolása ill. a sorozatunkban egy szám áthelyezése az üres helyre úgy történik, hogy közben páros sok helyet ugrunk át. A jellemzőben változást nyilván csak a mozgatott és az átugrott számok viszonyának megváltoztatásából nyerhetünk. Ha pl. előre (kisebb sorszámú helyre) ugrunk, akkor a jellemző egyrészt annyival csökken, ahány (az ugrónál) nagyobb sorszámú kockát ugortunk át, másrészt annyival nő, ahány kisebbet. De ha a kisebbek és a nagyobbak száma összesen páros szám, úgy különbségük is páros, tehát a jellemző valóban páros számmal változik. Hasonló a helyzet, ha hátrafelé ugrunk.

Állítás: ha mindent a helyén hagyunk, csak két kis kockát cserélünk ki, akkor a helyzet jellemzője páratlan számmal változik.

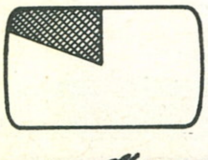
Indoklás: ha a két kocka szomszédos sorszámú helyen van, az állítás nyilvánvaló. Ha messzebb van, jobban meg kell gondolni a dolgot. A lényege az a dolognak, hogy a mindkettőnél kisebb számokat az egyik az egyik, a másik a másik irányba ugorja át, így összességében itt változás nem történik. Hasonló a helyzet a mindkettőnél nagyobb számokkal is. Ha pedig átugornak olyan számokat, melyek az egyiknél kisebbek, a másiknál nagyobbak, akkor az ezekkel való viszonyban a kisebbik által megtett ugrásokat is, és a nagyobbik által megtett (ellentétes irányú) ugráskor is ugyanannyi változás történik, így ez páros változást jelent. Végül itt is a két ugró kocka megcserélődése még jelent 1 változást, tehát a jellemző végül tényleg páratlan számmal változik.

A két állításból már következik eredeti állításunk, hisz ha egy számot mindig páros számmal változtatgatunk, a végén nem változhat páratlan számmal. Hát ez jó bonyolult lett! Biztos akadnak olvasóink, akik szívesebben látnák ezt matematikusabban, precízebben megfogalmazva (úgy kb. negyede ilyen hosszú lenne), de azt hisszük mégis többségben vannak, akiknek még ez is inkább túl matematikus. Sajnáljuk, de bizonyos fokú precízséget kénytelenek voltunk vinni a bizonyításba.



Múltkori pályázatunk sikerén felbuzdulva most ismét szakköri pályázatot indítunk, igaz ez a tanév vége miatt csak 2 hónapos lesz. Úgy gondoltuk, hogy most 2 könnyebb feladatot tűzünk ki (a díj is kisebb), hogy minden iskola kezdőbb szakkörei is benevezhessenek. Szeptemberben aztán majd újra indul egy nagyobb pályázat.

Most a díj: **1 finomgrafikai egység a HT-hez. A MICOLOR nevű készüléket tavaly októberi számunkban mutattuk be! A programokat most is kazettán kérjük beküldeni (legalább 3-szor felvéve, ahol lehetséges, ott különböző gépeken is) rövid leírással együtt a BIT-LET címére**



AZ 1. FELADAT

Készítsünk kaleidoszkópot! Tehát egy grafikai programot kérünk, a következő megkötésekkel:
1. **A program HT gépre készüljön BASIC nyelven, most gépi kódú rutinok használata nem engedélyezett!**

2. **A képernyő egyik nyolcadában (I. az ábrát!) minden állapotban legyen egy véletlenszerű minta, a többi nyolcadot ennek tükörképeivel töltse ki a gép.**
3. **Az állapotok ne úgy váltsák egymást, hogy először a „vezérnyolcad" változik, utána többi, hanem szép, szimmetrikus módon történjen a változás.**
Jó szórakozást!

Kérjük levégni és a levélre felragasztani!
Beküldési határidő május 15.

BIT-LET
Szerkesztőségünk címe ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál
Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Sorvezető: Székely Jenő. Posta. Halász Péter